EXECUTIVE SUMMARY

The Defense Information Infrastructure (DII) Common Operating Environment (COE) includes distributed computing services to provide specialized support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment. The commercial software selected by the DII to provide these services is the OSF's DCE. Implementation of DCE is planned for DII COE Version 3.0. The purpose of this report is to outline all of the activities that must occur for a successful implementation of DCE within the DII. The first system implemented under the DII COE will be the Global Command and Control System (GCCS).

The implementation of a distributed computing environment (DCE) is a complex task that requires deliberate planning. Decisions made during initial implementation of a global distributed computing environment may be difficult to change as the system expands and matures. Critical technical decisions include issues such as: which features of DCE to employ, cell organization, directory structure, and server placement. Administrative and policy decisions include assignment of management responsibilities, security policies, and training requirements. Decisions are also required on mandatory or desired coding practices. Once these decisions are made, additional effort is required to develop guidelines and scripts to actually effect an installation of DCE at the GCCS sites and for other DII sites.

The objectives for DCE in DII COE the 3.0 are fairly limited. DCE provides great opportunities for transparent, secure, distributed computing, but these capabilities do not come "out of the box"; they must be programmed into distributed applications. Applications cannot use these capabilities until the basic DCE services are installed and configured. Therefore, the objective for DCE in DII COE Version 3.0 is not a full-fledged conversion or development of all applications to DCE. Instead, the objectives are to establish the DCE mechanisms so that applications can be converted and developed in later releases. The following are the specific DCE objectives to be achieved in DII COE Version 3.0:

- **Establish the DCE Infrastructure**. Establish all of the crucial segments of the infrastructure including both the technology infrastructure of DCE services, and the administrative infrastructure of policies and people for the successful maintenance of the technology.

- **Establish an initial DII global file system prototype for GCCS**. The GCCS DCE Distributed File System (DFS) provides opportunities for transparent sharing of information on a world-wide basis. An initial capability will be installed in Version 3.0 using DFS servers at four sites to provide world-wide access to an initially-limited set of files.

- **Provide DCE guidance for developers**. It is assumed that there will be few applications developed or re-engineered to use the DCE remote procedure call (RPC) capabilities for Version 3.0. However, an important part of the DCE infrastructure to support future development is the establishment of guidelines on the use of DCE for the Developer's Toolkit. Tools to assist the developers will also be investigated and acquired. This objective does not need to be completed by the time Version 3.0 is installed, but it is considered part of the Version 3.0 effort because of its importance for future releases.

- **Establish a knowledge base for future evolution**. Perhaps the most important objective of Version 3.0 is to gain hands-on knowledge and experience with the use and administration of DCE. The DII COE when used by GCCS, with over 30 worldwide sites, will be one of the larger (if not the largest) DCE installations currently in existence. The other objectives for Version 3.0 are intentionally kept fairly modest to allow time for training, discovery, and adaptation based on the experience in this version.

The Implementation Plan includes a comprehensive set of actions required to achieve these objectives. The actions fall into the following strategic groupings:

- **Implement and Install Basic DCE Services**. This will provide access to the DCE services: directory service, security service, time service, and RPC. The DCE software will be a Commercial-off -the Shelf (COTS) product that will support the DII COE hardware platforms. The software must be configured for DII and installed on machines at each site. A major implementation decision involves the definition of DCE cells -- the basic administration unit for DCE. Guidance for DII cell structure is provided in Appendix B. However, each system like the Global Combat Support System (GCSS) and GCCS will require additional analysis to determine an appropriate DCE cell structure.

- **Establish DCE Administrative Infrastructure**. In addition to installing the DCE software, The system must establish a management and administration structure to maintain DCE: creating users, allocating services, and performing backups and restores. It is recommended that the responsibility for DCE administration be assigned to the system administrators at each site. Procedures must be developed and users trained. Experience gained from the initial implementation(s) will assist in developing an overall DCE administrative infrastructure for the DII.

- **Establish DCE Security Infrastructure**. Closely related to DCE administration is security administration. It is recommended that current security personnel at each site perform DCE security administration. Security administrators are responsible for assigning identifiers, groups, and maintenance of access control lists (ACL). As much as possible, DCE security should be integrated with existing DII and system security mechanisms. Policies must be

developed, procedures written, and administrators trained. These policies and procedures will be used as a baseline to develop an overall DII COE security architecture.

• **Establish Distributed File System**. The DFS provides a transparent, secure global file system. DFS has enormous potential for sharing files within and between sites. Initially DFS will be used for the GCCS community, but the usage will grow as experience is gained. Appendix D provides additional details. This prototype effort will be used as a template for developing a global (DFS) file system for the DII. DFS will be installed within a global cell that has machines at four sites world-wide (DISA, TRANSCOM, EUCOM, and PACOM). This cell will provide secure, global visibility to current information using automatic replication. All GCCS sites will share files by access to a file server within this cell.

• **Establish Migration Strategy for DII Applications**. As the technical and administrative infrastructure is being established, plans will be made for the migration of DII applications to DCE. A set of common tools and guidelines will be selected or developed to simplify the task of migrating DII applications. The goal will be to provide an open and extensible environment to reduce the time required for the migration and to maximize the commonality in approach across the DII application suites. Early systems will be used as the prototype for the DII migration strategy. The migration strategy will include appropriate training for system and DII COE developers. It is assumed that very few applications will actually use DCE in Version 3.0. This will allow time for guidelines to be developed and tools to be selected. The major movement to use DCE will occur in Version 4.0 and later versions of the DII COE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

The Defense Information Systems Agency (DISA), Joint Interoperability and Engineering Organization (JIEO) Center for Computer Systems Engineering (CFCSE) is responsible for the DII COE. This plan describes the implementation of DCE as part of the COE.

1.1 Background

The DII supports an open system environment for automated information processing users at all levels of the Department of Defense (DOD). The DII COE includes support applications, platform services, and reusable software components. The COE provides integrated support services that support the mission application software requirements and a software development environment to assist in the development of mission applications.

The DII COE includes distributed computing services to provide specialized support for applications that may be dispersed among computer systems in the network but must maintain a cooperative processing environment. The commercial software selected by the DII to provide these services is the Open Software Foundation's (OSF) DCE.

The OSF's DCE is not a single product, but an integrated set of software components that can provide for resource sharing, security, and a common time reference, allowing applications using those services to interoperate on a variety of platforms regardless of the manufacturer of the hardware, operating system, or network software. Applications can make use of one or more of the services provided by DCE without using them all.

1.2 Purpose

The implementation of DCE is a complex task that requires deliberate planning. Decisions made during initial implementation of a global distributed computing environment may be difficult to change as the system expands and matures. Critical technical decisions include issues such as: which features of DCE to employ, cell organization, directory structure, and server placement. Administrative and policy decisions include assignment of management responsibilities, security policies, and training requirements. Decisions are also required on mandatory or desired coding practices. Once these decisions are made, additional effort is required to develop guidelines and scripts to actually effect an installation of DCE at the sites. GCCS is the first systeplanning to imnplement DCE within the DII COE. As such, this implementation plan contains actions and guidance for the GCCS architecture as an example. Lessons learned from the GCCS implementation will provide an experience base for planning future DII COE implementations.

The purpose of this report is to outline all of the activities that must occur for a successful

implementation of DCE within the DII COE.

1.3 Report Organization

This report consists of six sections and eight appendices. They are:

- Section 1. Introduction - Provides a background of the DII DCE task, its purpose, and report organization.

- Section 2. Objectives - Describes the limited objectives for DCE in the DII COE Version 3.0. Although DCE has tremendous potential once it is implemented within applications, the advice of successful implementors of DCE is to move cautiously. DII applications can not use DCE until the basic DCE infrastructure services are in place; therefore, the primary objectives in Version 3.0 are to install these infrastructure services and gain experience and confidence in their use.

- Section 3. Strategy - Describes the overall approach or strategy for the use of DCE within DII COE Version 3.0.

- Section 4. Issues/Impacts - Describes the impacts of DCE on the DII COE in Version 3.0. Impacts are described from the point of view of: users, hardware, system and security administrators, and application developers. It also describes classes of tools that may be necessary for effective implementation of DCE.

- Section 5. Action Plan - Outlines the actions necessary for DCE implementation. Appendices provide draft versions of some of the decision papers and guidelines that must be developed to support the implementation. At this point, the action plan is primarily a checklist describing required actions. Later versions of this report should include assignment of responsibilities for these actions, estimates of resources required, and a timeline showing action dependencies and schedule.

- Section 6. Version 4.0 Plans and Issues - Provides an overview of expectations for the use of DCE in future releases of the DII COE.

- Appendix A - Acronyms

- Appendix B - Proposed Cell Structure

- Appendix C - Proposed Tools Analysis

- Appendix D - Proposed Distributed File System Utilization
- Appendix E - Proposed DCE Server Allocation
- Appendix F - GCCS DCE

2.0 OBJECTIVES

This section describes the objectives for DCE in DII COE Version 3.0, which are fairly limited. In a previous report documenting Lessons Learned in the use of DCE within commercial and DOD organizations, one of the important findings was that implementation should be undertaken in deliberate steps. DCE provides great opportunities for transparent, secure, distributed computing, but these capabilities do not come "out of the box"; they must be programmed into distributed applications. Applications cannot use these capabilities until the basic DCE services are installed and configured. Therefore, the objective for DCE in DII COE Version 3.0 is not a full-fledged conversion of all applications to DCE. Instead, the objectives are to establish the DCE mechanisms so that applications can be converted in later releases.

The following are the DCE objectives to be achieved in DII COE Version 3.0:

- **Establish the DCE Infrastructure**. This infrastructure includes both the technology infrastructure of DCE services, and also the administrative infrastructure of policies and people for the successful maintenance of the technology. Version 3.0 will establish this infrastructure.

- **Establish an initial GCCS global file system**. The DCE Distributed File System (DFS) provides opportunities for transparent sharing of information on a world-wide basis. An initial capability will be installed for GCCS using DFS servers at four sites to provide world-wide access to an initially-limited set of files. The GCCS global file system will be the prototype for other DII global file systems.

- **Provide DCE guidance for developers**. It is assumed that there will be few applications developed or re-engineered to use the DCE remote procedure call (RPC) capabilities for Version 3.0. However, an important part of the DCE infrastructure to support future development is the establishment of guidelines on the use of DCE for the Developer's Toolkit. Actual tools to assist the developers will also be investigated and acquired. This objective does not need to be completed by the time Version 3.0 is installed, but it is considered part of the Version 3.0 effort because of its importance for future releases.

- **Establish a knowledge base for future evolution**. Perhaps the most important objective of Version 3.0 is to gain hands-on knowledge and experience with the use and administration of DCE. The DII when used by GCCS, with over 30 world-wide sites, will be one of the larger DCE installations currently in existence. The other objectives for Version 3.0 are intentionally kept fairly modest to allow time for training, discovery, and adaptation based on the experience in this version.

3.0 STRATEGY

This section describes the overall strategy for achieving the objectives for Version 3.0. This strategy includes the elements listed below. These are listed as if they are sequential steps, but will proceed in parallel.

3.1 Implement and Install Basic DCE Services

First, the DCE software environment will be installed in each of the sites. This will provide access to the DCE services: directory service, security service, time service, and RPC. These basic services will also provide a framework for building distributed applications, establishing a unified security realm, and a common timing service.

The DCE software will be procured from a commercial resource and will be a Commercial-Off-the-Shelf (COTS) product. This software will support the different DII hardware platforms and operating systems. A major implementation decision involves the definition of DCE cells -- the basic administration unit for DCE. Guidance for defining the DII cell structure is provided in Appendix B. Specific guidance for establishing the GCCS cell structure is provided in Appendix F. Experienced gained from this initial implementation will be used to develop subsequent DII cell structures.

3.2 Establish DCE Administrative Infrastructure

In addition to installing the DCE software, a system must establish a management and administration structure to maintain DCE: creating users, allocating services, and performing backups and restores. It is recommended that the responsibility for DCE administration be assigned to the system administrators at each site. Procedures must be developed and users trained. Experienced gained from the implementation will be used to develop a DII administrative infrastructure.

3.3 Establish DCE Security Infrastructure

Closely related to DCE administration is security administration. This is described separately because it is recommended that current security personnel at each site perform security administration. Security administrators are responsible for assigning identifiers, groups, and maintenance of access control lists (ACL). As much as possible, DCE security should be integrated with existing system and DII COE security mechanisms. Policies must be developed, procedures written, and administrators trained. Experienced gained from the initial implementation will be used to develop a DII security infrastructure.

3.4 Establish Distributed File System

The previous steps only lay the foundation for developing distributed applications -- they have no effect until applications are developed to use this foundation. The DFS is a distributed application that is part of DCE and provides a transparent, secure global file system. DFS has enormous potential for sharing files within and between sites. However, DFS administration is complex and DFS management software is not as readily available. For these reasons, the implementation of DFS in GCCS is limited to a prototyping effort with limited use by GCCS applications.

3.5 Establish Migration Strategy for Major DII Applications

Lessons learned from the early system migration will be used to develop a DII migration strategy. As the technical and administrative infrastructure is being established, plans will be made for the migration of applications to DCE. A set of common tools and guidelines will be selected or developed to simplify the task of migrating applications. The goal will be to provide an open and extensible environment to reduce the time required for the migration and to maximize the commonality in approach across the legacy and DII COE application suites. The migration strategy will include appropriate training for system and DII COE developers. It is assumed that very few DII COE applications will actually use DCE in Version 3.0. This will allow time for guidelines to be developed and tools to be selected. The major movement to use DCE will occur in later as DOD systems migrate to the DII COE.

4.0 ISSUES/IMPACTS

This section describes the impact that DCE will have on DII COE Version 3.0 based systems from various points of view.

4.1 Administration

The biggest impact of DCE on the DII and system infrastructures will be on system and security administrators. The approach to implement the DCE infrastructure prior to migrating applications is designed to minimize this impact. It is assumed that existing system administrators and security administrators will take on the corresponding DCE roles. System administrators are responsible for overall setup, performance monitoring, and directory/file backup and restore. Security adiministrators are responsible for maintaining user and group identity lists, ACL and profiles.

In the initial implementation, all administration is performed by trained administrators, not by end users. This is possible because of the limited operational use of DCE. This may change in later releases as DCE use grows and some administrative responsibilities are delegated to user groups.

Administrative responsibilities include: initial installation and configuration, supporting DCE integration (creation of accounts, CDS directories, DFS filesets, etc.), and routine maintenance (DCE backups, load balancing).

DCE consists of a set of services: DFS, distributed time service (DTS), Security, RPC, and CDS. Several of these services maintain a database that require backup for use in case of catastophe. This is referred to as media backup. These same services require that a logical view of the services' database be maintained. The purpose of this data will be to reconstruct a database between backup versions.

Most DCE services are integrated with security. As such, security administration crosses into the administrative realm of DCE services. Tasks include defining initial ACLs and policies, and testing for compliance with security policies.

Most DCE services are replicatable and some are partionable. An administrator is responsible for ensuring that the replication strategy meets the criteria for safety and performance levels. This is a concern of DFS, DTS, Security, and CDS. The administrator can periodically monitor these services and make decisions as to the level of replication employed.

Several of the DCE services (DTS and RPC) have per-machine configuration settings. These services require special consideration because of the potentially distributed nature of a DCE cell. The same tasks described above apply, but on a larger scale. Fortunately, nearly all DCE

administration can be performed remotely. This allows the administrator to administer the entire cell from a single location.

Many initial configuration responsibilities have been simplified and based on the similarity between GCCS sites that allows site initial settings to be determined centrally and then tailored for each site at installation time. This may not necessarily be true for other DII systems.

4.2 Application Developers

It is assumed that there will be little application use of DCE RPC in Version 3.0. There is no impact on non-DCE applications and their developers.

There are several choices in terms of new development:

1) Educate developers on current DCE application programming interfaces (APIs). Develop application programs in 'C'or Ada. The disadvantage of this approach is the complexity of DCE and the lack of standardization.

2) Create a wrapper layer/library such that little or no DCE APIs are exposed other than Interface Definition Language (IDL). This deals with the complexity and standardization. This approach can be taken easily if support for C and Ada are the only required languages.

3) Adopt a third-party solution (e.g., Transarc's Encina) as a means of hiding the complexity and lack of standardization. Third-party solutions would be more costly, but eliminate the need to develop the wrapper/library described in 2) above. This approach may have additional benefits for migration to CORBA.

4) Adopt an object-oriented approach such as HP's Object-oriented DCE (OODCE). This is a major departure in that development would be performed using C++. Although no direct migration is possible to CORBA using this model, the object orientation is probably a very important migration step.

Each of these alternatives will be explored during the DII COE Version 3.0 timeframe. However, applications implemented in Version 3.0 should use the first or second approach.

4.3 Development Tools

The selection of development tools depends highly on the approach taken above in section 4.2. Development tools can be divided into the following areas:

1) Integration tools - tools which allow non-DCE environments to participate in DCE. Examples include Open Horizon's Connection product which integrates numerous clients and databases into DCE Security and CDS Services.

2) Development toosl which provide additional value to the DCE core services or facilitates the development of DCE-based applications. Tools available in this category include HP's OODCE for Solaris and HP, which provides an object-oriented environment for the development of distributed applications, and Transarc's Encina Toolkit and Services, which provides a set of DCE services/libraries for the development of robust or 3-tiered applications.

3) Development tools which provide assistance during the development cycle. Tools include compilers, debuggers, and other development aids. Examples include HP's distributed debugger, Purify from Pure Software, and others.

For the intial implementation of DCE in DII COEVersion 3.0, only some simple development tools will be provided along with the basic DCE software, other tools may be provided for later releases of the DII COE.

4.4 Management Tools

DCE Administration can be a very complex task even when administration is done properly. As reported in the Lessons Learned report, the management and administration tools that are included with the basic OSF DCE product are not very effective. Many DCE users have acquired or built additional administrative tools. In the initial implementation of DCE, there will not be widespread use of DCE by applications, so the administration should be limited to user administration, so the full impact of the complexity of DCE will not be felt. The benefits of a graphical administrative tool are that it reduces the time to learn and administer DCE environments.

The DCE software will provide a nominal set of administrative tools but most of administration will be accomplish via the command line. There are a limited number of graphical administrative tools available in the commercial environment. The DCE Working Group has recommended that the graphical administrative tool, Chisholm Cell Manager, be procured to assist in DCE administration. The strategy is for sites to use the administrative capabilities inherent in DCE. Sites have the option of procuring Chisholm Cell Manager for their local use. Chisholm Cell Manager will be segmented and available in the COE, sites only need to procure the appropriate license(s). The Army has developed several administrative tools as part of their DCE implementation and every effort will be made to leverage off these efforts.

The benefit of commercial tools is that they provide a nearly complete point-and-click or drag-and-drop graphical interface on top of DCE command-line administrative tools. DCE itself provides a powerful and extensible scripting language for administration (**dcecp**) which is accessible using command-line interfaces.

Today's administrative tools all suffer from three maladies: they are not task oriented, none support DFS administration, and with the exception of one, and are non-extensible. The last point is by far the most critical. With the exception of the vendor offerings, none of the administration tools provide DCE 1.1 capability. The good news is that even the vendor-provided tools are capable of managing other non-vendor DCE machines (with the exception of initial installation and configuration). Appendix C describes several tools which were considered.

Of these tools, Chisholm's Cell Manager is the most robust and comprehensive. This tool will be segmented for DII COE v3.0 but it will be provided as an optional tool. This means that the segment will not be loaded automatically and systems/sites will need to obtain their own licenses if they intend to use this product.

5.0 ACTION PLAN

This section identifies all of the actions required to put DCE in place for Version 3.0, following the strategy outlined in Section 3. Each action is described in a few sentences that identify what the action is, and why the action is needed. Preliminary solutions to some planning actions are included as appendices. There are specific actions that are required for the implementation of DCE for GCCS. These actions and guidance are provided in Appendix F.

Each of the strategic elements from Section 3 is decomposed into specific required actions. The actions within each subsection are grouped into five life-cycle phases:

- **Requirements analysis and design** - Actions required to reach policy or design decisions early in the implementation. Most of the appendicies are decision papers leading to solutions to these type of actions.

- **Development and test** - The actual development of products necessary to the implementation based on the design decisions above. Products include guidance documents, software, or installaction instructions and scripts. These actions also include exploratory testing by DISA and other development sites, as well as testing conducted by development activities.

- **Integration** - DISA will pull together the components and procedures developed above to create the final products to be delivered to operational sites. Also includes integration testing activities.

- **Installation** - After integration by DISA, DCE policies, procedures, and software must be installed at each operational site. These actions also include tailoring procedures to the individual site.

- **Operation** - These actions include all on-going activities that are required centrally and at each site to manage and maintain the DCE infrastructure.

5.1 Implement and Install Basic DCE Services

This section identifies all technical actions associated with developing, configuring, and installing the basic DCE software. It is not concerned with organizational, procedural, or personnel issues associated with administration or security, which are included in sections 5.2 and 5.3.

5.1.1 Requirements Analysis/Design

These actions are primarily initial policy and design decisions.

5.1.1.1 Determine Cell boundaries, names, and hierarchy

One of the most important issues that must be addressed prior to implementation is determining the cell structure. There are several criteria for determining a cell structure/architecture. A cell is a group of users, systems and resources that typically have a common purpose and share common DCE services. They should share a logical connection, functional relationship, or group association. Cell structure decisions are based on balancing the administrative load. The smaller the number of cells, the less infrastructure that is required for administration. If you have numerous small cells, then more administrative infrastructure is required. Conversely, a single cell will require less administrative infrastructure but will require one administrative center to do a tremendous amount of administrative overhead. Additionally, cell structure should be based on the requirement for common purpose and shared resources. Aside from administrative overhead, there are several factors that must be considered including performance, efficiency, and reliability.

For GCCS, the number of cells is based on the administrative structure of the 37 Initial Operating Capability (IOC) sites. The rationale for using the IOC is presented in detail in Appendix F. While this infrastructure appears to be the best solution for GCCS, experience gained in the engineering of the GCCS cell structure will be used to provide guidance for the other systems within the DII. What is appropriate for GCCS may not work well in other systems like GCSS. The GCSS organizational structure is not as well defined as GCCS and the integration of the different GCSS systems has yet to be engineered. GCCS has both a well defined organizational structure and a network baseline. Basic guidance for DII cell structure is provided in Appendix B.

5.1.1.2 Determine Initial Directory Naming

Identify the standard directory structure (for service names, application segments, profiles, etc.). Postulate additional work required when full-scale application development begins. Because few applications will make use of the CDS at furst, this activity can be limited. DFS directory and file names are discussed in Section 5.4. These guidelines will create directories for use by organizations, agencies, services, etc.

Implementation Note: *This task was completed and documented in Appendix X of the I&RTS, December 1995.*

5.1.2 Development/Test

Includes all software or procedure development and test actions. Also includes initial prototype

testing at the OSF and other sites.

5.1.2.1 Integrate DCE login with other COE logins

The goal is a single login and a single UNIX/DCE identity. Single login means that from the user's perspective, they perform a single sign-on regardless of their point of login. Integrated system/network security means that the user has a single identity across systems and DCE services. Single signon/login will be accomplished for DII COE Version 3.0.

5.1.2.2 Determine procedures for testing DCE COE

DCE should be installed at the OSF and used during COE development. Initial testing is designed to gain experience, not to test functionality. It is strongly recommended that DCE also be installed at other development sites and tested across the NIPRNET or SIPRNET. This will allow DII COE developers to gain experience with multi-site operation as procedures are being developed.

5.1.2.3 Develop site guidelines for server allocations

DCE includes 5 to 10 standard servers (daemons) and databases (e.g., CDS clearinghouse, DTS, security registry, etc.). For reliability and performance, each server may be resident on one or more machines at each site. Although the hardware configurations are similar, each site may be somewhat different. These guidelines, to be included with the installation instructions, will describe a standard client/server laydown, and site options depending on actual configuration.

Basic guidelines should include a replicated server for each primary database server (Security, CDS, FLDB). The security server should be placed on a physically secure machine which should preferably be dedicated to this purpose (and disallow login). DTS requires a minimum of three DTS servers. These servers should be selected on machines which are normally up 24 hours a day.

Appendix E contains a proposed generic server allocation and consolidated list of guidance areas. Appendix F contains a proposed server allocation for GCCS.

***Implementation Note**: This task was partially completed and documented in the DII COE DCE Administratin Guide, May 1996.*

5.1.2.4 Develop installation instructions and scripts

These should be step-by-step procedures for installing DCE at a site. Since all of the GCCS sites

are similar, standard scripts can be developed to automate much of the installation. This may not be true for other DII sites. The procedures will include instructions for modifying and running the scripts at the site. An installation and configuration log will be provided in an electronic or paper form for tracking the initial and on-going configuration actions that are performed. Installation procedures will include security related actions described in Section 5.3.

Implementation Note*: DCE client and server segments have been developed for version 2.0 and will be refined for version 3.0 A DFS segment will also be developed for version 3.0*

5.1.3 Integration

Includes actions required to prepare for delivery, including training.

5.1.3.1 Revise Integration Test procedures to include specific DCE tests

Because DCE is a major new system-level component, specific tests should be included in the integration testing to ensure that it is providing the proper COE support prior to integration and release.

5.1.3.2 Include DCE segment in DII COE delivery to development sites

Include DCE Version 1.1 in the stable COE delivery to DII software design activities so that it can be used in development and testing. Because it will be a major component of system security and integration, the DCE client should be part of the DII kernel while the DCE server should be a separate segment.

Implementation Note: *DCE client was implemented in the COE version 2.0. kernel while the DCE server was implemented as a separate segment.*

5.1.3.3 Include DCE segment in DII COE 3.0 final delivery to operational sites

Include DCE Version 1.1 in the delivery to operational sites.

5.1.4 Installation

The following actions must be performed at each development and operational site.

5.1.4.1 Determine machines to run each DCE component

Section 5.1.2.3 discusses this topic. The guidelines from Appendix E should be applied to

determine which machine(s) will run each service/database (e.g., CDS clearinghouse, DTS, security registry, etc.).

5.1.4.2 Install DCE software at each site

Install DCE software on clients and servers. The DCE client segment is part of the COE kernel. The server segment is a separate segment (not part of the kernel) since the server software will only be installed on a limited number of systems. The client installation is accomplished as part of COE software installation.

5.1.4.3 Configure DCE at each site

These are actions required at any DCE site within the DII. The COEInstaller will be used to install DCE at each of the sites. Segment installation and administrative script files will be designed to prompt the site for specific configuration parameters.

Security configuration is an essential portion of DCE configuration. All security related actions are consolidated in Section 5.3.

DCE automatically populates the CDS namespace with entries that it requires for normal operation. In order to support structured, controlled, and manageable access to CDS, a set of procedures will be developed on partitioning the CDS database and populating it with initial directories. DCE servers will use the configuration database provided by the **dced** service when they initialize. Servers developed for DII systems will use the 1.1 **dced** configuration database and thus will not register themselves directly. This indirection allows the configuration information to be remotely manageable and easily configurable.

Client machines can be added during installation or at a later time. The Army has developed as part of its DCE implementation, a tool for installing and de-installing clients. This tool was developed by UNIXPROS and is called the Cell Factory. This is one of the Army projects that will be considered for leverging into the DII COE.

For machines using the DTS time service, DTS client daemons will be configured. The use of DTS will not be mandated for all client machines providing that they are using some other reliable means of time synchronization. DCE requires that synchronization between machines throughout the DII be kept with five minute tolerance.

5.1.4.4 Install cell names in DNS

It is assumed that host names and address information have already been installed in the Domain

Name Service (DNS). Until hierarchical cells are established (see Appendix B), each cell needs to have cell information registered in DNS. This information cannot be obtained until these sites have been installed.

5.1.5 Operation

All operational activities are included in sections 5.2.5 and 5.3.5.

5.2 Establish DCE Administrative Infrastructure

This section identifies all actions associated with developing policies and procedures for DCE administration, organizing and training for administration, and configuring and installing the basic administration information. It is not concerned with security, which is included in a later section.

5.2.1 Requirements Analysis/Design

These actions are primarily initial policy and design decisions.

5.2.1.1 Determine administrative policies and responsibilities

Determine which office or person performs administrative roles, including security administration. Since DCE allows remote administration, describe the role of central site vs CINC/service sites. This decision must trade off the efficiencies of central administration against the autonomy of the sites and reliability.

Implementation Note: *This task was partially completed and documented in the DII COE DCE Administration Guide, May 1996.*

5.2.1.2 Determine requirements for additional administration tools

Section 4.6 discusses administrative tools and the potential need for tools beyond those that are included in basic DCE. A set of criteria should be developed for assessing the need and desirability of particular tools. Criteria should include: administration (security, **dced**), extensibility, ease of use, DFS administration, high-level task oriented operation (e.g., add users, security operations using extended attributes, backup, etc.), and cross-cell operation. The Distributed Computing Working Group (DCWG) recommended that the DCE Cell Manager from Chisholm be used to assist in Cell Administration. Cell Manager is the only GUI based software currently available for cell administration. It is currently under evaluation and will most likely be implemented as an option for DII COE sites. The software will be segmented and distributed with the COE but each site (cell) that wants to use it must purchase its own license.

Implementation Note: *This task was accomplished on July 30, 1996.*

5.2.1.3 Determine requirements for training

The previous Lessons Learned report stressed the importance of training to a successful DCE implementation. The following types of people require training:

1) Security administrators
2) Cell administrators (system administrators)
3) Application designers/developers
4) OSF/Help Desk or Theatre - Level 2 Support (DFS responsibility)
5) DCE Users.

The following curriculum or equivalent will be required for each role:

Table 5-1. DCE Training Requirements

	Security administrators	System administrators	Application developers	OSF/ Help Desk	DCE Users
Introduction to DCE - what it is and what it does (1 day)	X	X	X	X	X
Basic DCE Administration - principals of administration using the selected administrative tool suite (3 days)	X	X		X	
DCE Security Administration - managing and maintaining security on DCE 1.1 (2 days)	X	X		X	
DFS Administration - installing and managing DFS (3 days)		X		X	

5.2.2 Development/Test

Includes all software or procedure development and test actions. Also includes initial prototype testing at the OSF and other sites.

5.2.2.1 Develop procedures for CDS and DCE security, dced database backup/recovery

Provide step-by-step instructions for performing a backup or recovery of DCE administration and security databases. Include guidelines for frequency of performing backups. These instructions should be part of the administration training discussed elsewhere. If The DII COE selects an extensible administration tool, these capabilities should be included.

5.2.2.2 Develop procedures for change and problem management

Provide a set of procedures for recording configuration changes made to the DCE environment. This information should be kept in a designated directory in DFS as well as in a log book or an electronic reporting facility. Entries should be made for significant events such as adding clients, creating CDS/Security directories, creating extended **dced** or security attributes, **dced** configuration records, etc.

A means of tracking problems and their resolutions should be provided. Access to this information will be important for other administrators and for timely problem resolution. This information should be maintained and indexed within DFS for access by support teams.

5.2.2.3 Develop guidelines for defining site-unique groups / organizations

A standard set of groups will be defined for normal GCCS functions and services, but sites may have the need to define additional groups. The guidelines will provide guidance on how and when to define additional groups to prevent group conflicts. It must take into account the trust hierarchy established by the GCCS cell definition. These groups may be different for other DII systems like GCSS.

5.2.2.4 Acquire administrative tools

The administrative tools selected in 5.2.1.2 must be acquired. Some of the tools being considered are available at no cost; others are commercial tools that must be purchased. As noted in 5.2.1.2, the Cell Manager software will probably be available for DII COE Version 3.0. Acquisition of Cell Manager will be a system/site option.

5.2.2.5 Arrange for administrator training

Of all the training discussed in 5.2.1.4, administration training is the most critical. The training plan for GCCS calls for generic system training to be provided by the Air Force (AETC) who has been designated the single agency manager for all GCCS training. Applications training is supposed to be covered for the first year by the sponsoring service or agency and then training

would be transferred to the Air Force.

It is recommended that DCE follow the GCCS training policy in that DCE should be taught commercially for the initial fielding and then training would be transferred to AETC. This is the current training policy for applications that are accepted as part of the GCCS baseline.

It is also recommended that all administrator training be conducted as hands on training. It is recommended that DCE administrators have a working knowledge of UNIX and system administration for UNIX systems. These recommendations should be included in a structured training plan.

5.2.3 Integration

Includes actions required to prepare for delivery, including training.

5.2.3.1 Publish administrative procedures

The procedures and policies developed in 5.2.1.1 need to be coordinated with the sites and sent in advance. Once DFS is established, it provides an excellent mechanism for dissemination of procedures, recording problems, and distributing application segments.

5.2.3.2 Integrate administrative tools into DII COE Version 3.0 release

If additional tools are required, they must be integrated into the DII COE release. The available administrative tools may only include DCE tools (e.g., **cdscp**) at this time.

5.2.4 Installation

The following activities must be performed at each development and operational site.

5.2.4.1 Install administrative tools

This is assumed to be part of the basic DII COE release.

5.2.4.2 Establish principals and groups

Overlaps exist between the roles of system administrator and security administrator. For example, principles and groups are required for locating DCE services, but they are also key to DCE access control. The following are described as systems administration actions, but require coordination with the security administrator.

During installation, DCE establishes a default *cell_admin* identity to perform cell administration. However, administrators should have their own accounts since using the default *cell_admin* leads to a lack of accountability. Credentials for administrators must be established as part of initial installation.

Once they are established, the system or security administrator can set up the remaining groups and users. Migration procedures should be defined for taking existing information (password files, security profiles, etc.) and migrating this into the DCE Registry. The selection of groups and principles are discussed in section 5.3.

5.2.4.3 Establish profiles for initial services

Profiles are used to define the search procedure for binding a client request to a service. It is primarily required when there are alternate servers available for a service (as there are with most of the basic DCE services). The profiles can be used to create a geographical hierarchy wherein sites prefer to use services within their own cell, then theatre, then any available service. By creating a profile of profiles, the first level profiles would differ by geography and the second level profiles could be shared. This activity can be deferred until after the initial DCE implementation because it is only relevant to server applications that use DCE RPC.

The Army has developed a publish-and-subscribe mechanism that should also be considered. This mechanism will need to be converted to work with TRANSARC DCE. This mechanism could be useful but it is important to understand that each site may be configured slightly different which may have an adverse effect on automating this process.

5.2.5 Operation

Includes on-going actions and activities at each site or at a central administration site.

5.2.5.1 Make Changes to the DCE Environment

Standard procedures will be defined using the selected tools including change management.

In a typical DCE environment, normal users and applications developers do not have modification access to the production cell. Changes are envisioned as being motivated by the installation of a new application segment, a new GCCS/DII COE release, load balancing, and equipment/role shuffling.

Changes made by the administrator should be logged in a change management system. For GCCS, an electronic form will be placed in a dated file in a DFS directory unique to each site.

Changes should also be accompanied by some documentation (e-mail, problem ticket, etc.). Changes must also meet the requirements of the Naming Procedures for services requiring them. Examples include selection of names or creation of directories in Security, adding or changing a directory in CDS, or creating or modifying a new DFS fileset.

5.2.5.2 Perform DCE maintenance

Maintenance covers tasks involving the ongoing support of the DCE environment, rather than tasks associated with the changes required to the DCE environment (e.g., adding users, configuring equipment, etc.). The procedures for maintaining DCE are documented in the DCE Cell Administration Guide which has been developed in Draft.

5.3 Establish DCE Security Infrastructure

This section identifies all actions associated with developing policies and procedures for DCE security, organizing and training for security administration, and configuring and installing the basic security information.

5.3.1 Requirements Analysis/Design

The purpose of the security infrastructure is to create an environment with controllable and accountable access to resources. The DCE infrastructure encompasses the identification of all security principals, authorization of all DCE resources and facilities for integrating other security systems.

5.3.1.1 Determine initial security policies

Security policies affect the installation and operation of DCE security parameters and functions. A set of policy decisions relevant to DCE must be developed prior to the deployment of DCE. The DCE security registry also maintains a set of policies and properties.

The DCE Security Registry uses privilege attributes to make authorization decisions. These attributes are referred to as Principals, Groups, and Organizations (PGO).

***Implementation Note**: The DCE Cell Administration Guide provides detailed information on setting up security for a DII DCE Cell. The Administration Guide covers security policies and properties, the PGO infrastructure, foreign cell trust and other items required to fully configure Cell security.*

5.3.1.2 Determine requirements for additional security tools

No Commercial off-the-shelf (COTS) tools exist today for security management in a DCE environment. Potentially needed tools include audit reporting, automated audit trail management, reduction, and archiving. Security alert monitoring, ACL auditing, and general security checklist applications would be beneficial.

Integration of non-DCE applications and Databases into the Security framework is important. Only Version 3.0 security requirements will be included. These may be minimal.

A mechanism should be developed for assuring that the ACLs for DCE objects are within the prototype ACL definitions defined by the security policy. (Basically an ACL audit).

5.3.1.3 Determine requirements and approach for security training

See discussion in Section 5.2.1.3 above.

5.3.2 Development/Test

Includes all software or procedure development and test actions. Also includes initial prototype testing at the OSF and other sites.

5.3.2.1 Develop software to implement password policies

As delivered, DCE allows the user to change their own password as frequently or infrequently as desired, and does not check for common words as passwords. DCE Version 1.1 contains security policies and properties that exercise limited control of password content (e.g., aging, content, etc.) These policies and properties have been aligned with the GCCS security policy and may be found in the DII COE DCE Cell Administration Guide. Since the goal is to have a single signon, password checking will most likely be done in CDE. This means that password checking must also be accomplished when a user changes his password (in CDE) and the resulting password must be used to manually update the DCE password.

5.3.2.2 Select/acquire/develop security tools

Implement any tools selected in 5.3.1.2.

5.3.2.3 Identify source for security training and arrange for training

The following companies may offer off-the-shelf and custom training for DCE Security: IntelliSoft, Curriculum, University of MN, Transarc, HP, IBM, and OSF.

5.3.3 Integration

Includes actions required to prepare for delivery, including training.

TBD

5.3.4 Installation

The following activities must be performed at each development and operational site.

5.3.4.1 Install DCE security software

Included as part of DCE installation.

5.3.4.2 Train security administrators for each site

See section 5.2.1.3.

5.3.4.3 Configure Security server

The security server is the first server created in a DCE cell. The system administrator, with the security administrator, will configure the primary and any secondary security servers. He/she will select the initial database and security-administrator's password, create the common security names and directories, set the registry properties and policies, and set the initial ACLs. These steps will be defined in the DCE installation procedure.

5.3.4.4 Establish ACLs for basic services

ACLs need to be established so that as new elements are added to the security space (database), the appropriate protections are applied. There may also be reason to correct the initial security settings that are shipped with DCE from the supplier.

5.3.5 Operation

5.3.5.1 Maintaining the security registry

These tasks will be performed by the Security Administrator (based on the assumption that the System Administrator doesn't have the permission to perform them). See the DCE Cell Administration Guide for more detail.

Maintenance of the Registry involves the following tasks:

1) Backing up the Registry.
2) Changing the Registry Password.
3) Monitoring the status of Registry replicas (additional servers).
4) Collecting and pruning the Audit Records.
5) Auditing permissions (ACLs) on DCE Objects.

5.3.5.2 Maintain ACLs as people/services come and go

There are several security related tasks involved in transitioning new users, equipment, or applications (servers). The actual settings for these ACLs are based on the type of operation being performed. The following general scenarios exist:

1) Users' private resources (things a new user might control) - examples might include a directory in CDS or DFS. A user is typically given explicit ownership of these resources or the service might grant the owner of a resource implicit control (a file server).

2) Users' access to other resources (things a new user might need to do his/her job). Users are typically granted access by means of group ACLs to servers or resources as needed. Using groups, an application typically define a user community and an administrative community, or an untrusted user set and a trusted user set. An attempt should be made to use only as much granularity as is required.

3) ACLs representing control over equipment (determines who has control of the equipment). ACLs on new equipment are generally used to determine those permitted to perform administration. Examples include: performing backup or managing the time.

4) ACLs used internally for application functioning (e.g., multiple cooperating servers). These ACLs are used by several DCE and like services that are composed of multiple servers. Today each of these servers (i.e., those running on different machines) use ACLs which cooperate (e.g., replication).

5.4 Establish DFS Infrastructure

Develop requirements and guidelines for implementing DFS.

5.4.1 Analysis/Design

DFS has advantages, but also imposes an administration burden. The DC Working Group

(DCWG) has developed a survey that could help decide how to implement DFS. The recommendation is to implement a scaled down deployment of DFS which provides access, but minimizes the administrative impact.

5.4.1.1 Determine how DFS will be used in DII Systems

A discussion of how DFS could initially be used in GCCS is located in Appendix D. Other DII systems could use the same strategy.

5.4.1.2 Determine DFS test requirements and approach

Like the rest of DCE, DFS should be installed at the OSF and used during COE development. Initial testing is designed to gain experience, not to test functionality. Since the global DFS cell is relatively autonomous from the rest of GCCS, it could even be installed in advance of the DCE implementation for wide-spread prototype usage.

5.4.1.3 Select files to install in DFS

The following types of files are the most obvious candidates for DFS:

1) Files that have broad geographic usage. Files that are accessed for read mostly stand to gain the most.

2) Files that are accessed using release control methodology (versioning).

3) Files accessed by individuals who are mobile.

4) Files whose access should be protected across the entire DII.

System users and application developers should be consulted to determine the specific groups of files to install in DFS.

5.4.1.4 Determine initial DFS directory/file structure

Since the DFS is global, it is important to get the structure right at the beginning. Again, choice of high level directories, permissions on those directories, the type of DFS mount points, and fileset names require planning. An initial proposal will be developed in Appendix D.

The structure of the hierarchy has implications on security and access type. The general rule is to place more secure data lower in the file tree, limiting access to all but a few upper directories.

The position in the tree where the mount points are created also has an impact for replicated file systems, and DFS has special algorithms for selecting a read-only or read-write version.

Based on the candidate files identified above, a recommendation will be made as to the logical groupings of these files. For example, within an application segment, files may be categorized based on their backup requirements, their modification pattern (frequent, infrequent, never), or growth patterns.

The names of the filesets also have an impact on certain operations. For example, filesets can be managed in a group if they have a common extension, such that *.src might include the filesets which need to be backed up daily.

5.4.1.5 Determine servers to house DFS and DFS gateways

The recommended approach in Appendix D will install DFS on new dedicated servers. If this is not possible, then the approach must be reviewed and a decision made on where to host DFS. If the proposed Network File System (NFS)-to-DFS gateway is accepted then a decision must also be made on which machine to use as a gateway at each site. Appendix E proposes that one of the applications servers be used.

5.4.1.6 Assign DFS administration responsibilities

As in the other services, DFS places requirements on both system and security administration. Administrative responsibilities for DFS clients are minimal or none if the initial configuration is performed properly.

The global cell will manage the DFS server equipment in each of the theaters. The implications are that certain administrative operations, such as backup, will only be performed at the theater sites. Giving a theater-level administrator these responsibilities and security control over some theater-local filesets seems reasonable.

5.4.1.7 Poll user community for requirements

After the system users have gained some experience using the initial global DFS, a survey should be conducted to determine how or whether to expand the use of DFS in later releases. A primary result of this should be to find out if DFS is required within each site. Similar analysis should be conducted to determine DFS requirements for other DII systems like GCSS.

5.4.2 Development/Test

5.4.2.1 Acquire DFS software

The DCE software and DFS software will be procured from the Transarc Corporation. While DFS server software will only be installed at the four primary node sites, access to DFS will depend upon the DCE implementation schedule. If the prototype DFS architecture is in place prior to DCE implementation, NFS-to-DFS gateway software will be provided on each of the DFS servers in order for non-DCE users to access DFS. If DCE is implemented prior to DFS than each user will have DFS client software and will be able to access DFS directly without using a gateway.

5.4.2.2 Conduct prototype testing at OSF

The DFS software should be installed and tested at the OSF and other sites to test and verify the integration of DCE software, DFS, and the gateway software. The experience gained should be included in the guidelines published.

5.4.2.3 Develop DFS administration procedures

Provide step-by-step instructions for performing a backup or recovery of DFS administration and filesets databases. Include guidelines for frequency of performing backups. These instructions should be part of the administration training discussed in Section 5.2.1.4.

These procedures also include steps for adding new filesets, mounting filesets in the DFS file tree, and performing load balancing, and storage management.

5.4.2.4 Develop installation scripts/procedures

Some or all of the files that will be moved into DFS will already be resident on the GCCS UNIX systems. It is desirable to make the DFS files accessible through their former UNIX file names, to reduce application impact. Since the GCCS sites are similar, standard scripts can make all of these changes, subject to some local customization.

5.4.2.4.1 Establish soft links for DFS directories in UNIX file system

DFS as a whole lives in "`/...`". For example "`/.../cellname/filename`" is the nomenclature used to reference a DFS file. Note that "`cellname`" would be the name of the global cell initially. It is perfectly valid to create a symbolic link (e.g., `/gccs`) on every DFS client machine. This could also be used to create soft link aliases for files moved into DFS, so that file names do not have to change immediately within the system scripts.

5.4.2.4.2 Build ACLs for files

Control of file resources is done using ACLs. Directories managed by sites or projects will maintain administrative control of these directories (at the top of a fileset). Common fileset directories will be owned by the global cell with read permission as appropriate.

5.4.2.4.3 Build Administrative Lists

Administrative lists will not be used as part of the initial implementation of DCE. Initially, access to the DFS file structure will be controlled using access control lists (ACLs). Administrative lists can be implemented at a later date if more granularity is required for access control.

5.4.3 Integration

Includes actions required to prepare for delivery, including training.

5.4.3.1 Install DFS as part of COE

Although the DCE client software should be part of the COE kernel, DFS should not be included in the kernel. DFS will be installed only on servers that are not part of the normal DII installation. The NFS-to-DFS gateway software should be part of the COE release, but do not need to be part of the kernel. It will be available as a segment.

5.4.3.2 Train DFS administrators

Using the approach outlined in Appendix D, only theater-level administrators will manage DFS servers. In addition to the system administration courses already recommended in 5.2.1.3, these administrators should attend DFS Administration - installing and managing DFS (3 days).

5.4.4 Installation

The following activities must be performed at each development and operational site.

5.4.4.1 Install and configure DFS software

Install DFS at the four GCCS global cell sites using published instructions. NFS-to-DFS gateway software will be installed at each site as part of normal DII COE installation.

5.4.4.2 Install/move files into DFS

At the primary GCCS global site, load most of the read-mostly filesets into DFS in advance. During installation at each GCCS site, migrate any site-unique files that are to be moved into DFS. Also delete (or rename) any local files that are now present in DFS.

5.4.4.3 Set up ACLs for local groups/principals

Create appropriate ACLs for any site-local groups and principals. The use of principal ACLs should be limited to filesets owned by individuals (such as home directories) (post 3.0). Normally access is controlled using security groups.

5.4.5 Operation

5.4.5.1 Perform DFS administration

As stated earlier, administrative responsibilities will lie with a trained administrator at a site in each theater.

The DFS administration activities include:

1) Backup/Restore operations (physical media).
2) Backup of the logical DFS state (**FLDB**). Only performed by the central site.
3) Management/monitoring of DFS storage.
4) Periodic replication of filesets as necessary.
5) ACL management.
6) Server load balancing.

5.4.5.1.1 DFS backup/restores

DFS backup is similar to backup in UNIX with the following exceptions:

1) DFS supports live backup (file systems don't need to be unmounted or left unaccessed).

2) DFS supports a 'backup fileset' which maintains a previous copy of a fileset in an efficient manner. This allows users an easy, efficient, and on-line means of retrieving the previous week's effort.

Guidelines will be developed indicating the frequency of backup and creation of backup filesets based on the type of fileset (indicated by its name). For example: `.src` filesets may be backed up daily with a backup fileset being produced weekly.

5.4.5.1.2 DFS tuning (moving filesets)

DFS allows an administrator to move filesets without disturbing active clients. This becomes necessary when a DFS aggregate (like a partition) becomes full or near-full or when the load on a file server become unbalanced. These should not be immediate concerns because we will have a good idea of how much data is being placed in DFS; the amount of initial access is quite low. Other tuneables exist at the file server and file client machines, such as the number of daemons, size of the cache, and communication chunk size.

5.4.5.1.3 Maintain DFS ACLs

This task involves checking that ACLs have not been tampered with and that compliance with DII COE recommendations is ensured.

5.4.5.2 Install other files into DFS as required

As popularity and publicity of the facility spreads, more applications and users will find value in using DFS. Users normally can't deposit arbitrary data in the DFS filesystem unless they are using a fileset that houses their private data or when collaborating on a shared project.

5.5 Establish Distributed Application Structure

This is an initial structure for developing DCE applications.

5.5.1 Analysis/Design

This document proposes a set of guidelines for how a distributed application using DCE services and resources. It does not address how existing applications should be converted into distributed applications, nor the use of specific programming styles. It does address several important DCE facilities which relate to the manageability of a DCE application in a large environment.

5.5.1.1 Determine the initial CDS directory structure for registering applications

DCE applications use CDS for two main purposes: servers record their location information (called binding information), which clients use to locate appropriate servers, and applications can use CDS to catalog certain resources or attributes which they possess. It is important that the use of CDS be structured because elements in CDS do not contain any explicit ownership information. CDS employs standard DCE ACL security which is used to control access to the directory.

Each cell has its own private namespace but each cell's namespace is accessible using a global name. For example, `/.../gccs.mil/pacom/applications/jopes` is a directory at the PACOM site being used for the JOPES application whereas `/.../gccs.mil/eucom/applications/jopes` is a directory in EUCOM. By having a common scheme of structuring CDS, a much higher probability of successful management and application portability can be achieved.

The approach to managing the CDS namespace is to use separate directory subtrees for each application segment. This allows the choice of directory names to indicate ownership and or function. Each site will have a top-level directory `/.:/applications` under which any system S application can create a subdirectory (e.g., `jopes`). Under the application directory will be directories for each of the type of CDS entries: profile, group, servers, and objects. Applications are free to define any directory structure under their application as needed.

Another benefit of this approach is that each application directory can be owned (from a security perspective) by the group which is responsible for the application.

Implementation Note*: The design of the CDS namespace for the DII was completed and documented in the I&RTS Appendix X, December 1996.*

5.5.1.2 Determine preferred application interfaces and tools

The DII COE is currently supporting Sun and HP computers as servers. For DII COE 3.0, the list of servers may grow to DEC Alpha, IBM AIX, and Silicon Graphics IRIX. On the client side, Sun, HP, DEC, IBM and Windows NT will be supported. A decision will have to be made as to which (either or both) platforms will be used for development of the DCE applications. Based on this decision, a set of development tools can be selected. Not all tools are available for both platforms and each has its own set of strengths.

Developers will be responsible for their own DCE training. DISA will be providing some limited training on the DII COE and has provided DCE guidance in Appendix X of the I&RTS.

5.5.1.2.1 Develop requirements/criteria for selection

There are very few tools available for DCE application developers and even fewer commercial interface packages. Interface providers supply layered products which add capabilities to the DCE applications. Tools provide benefits during the development cycle. Examples include integrated development solutions, incremental compilers, graphical debuggers, performance tuning, etc.

The following criteria should be considered in evaluating development environments:

- Support for DCE threads
- Cross-vendor support
- Distributed debugging capability
- Run-time checking (memory etc.)
- DCE protocol analysis.
- Ada Support

The following criteria should be considered in evaluating DCE support libraries:

- Ability to extend/modify
- Ability to participate in an eventual migration to CORBA
- Completeness of support functions
- Use of DCE 1.1 capabilities (rather than native implementations).

5.5.1.2.2 Examine available techniques and tools

Transarc and other vendors distribute the Encina Toolkit which provides a layer of service above the DCE including initialization, load balancing, and server management. The other candidate for this function is to develop or contract development of a set of interfaces which are ideally suited to GCCS.

Pure Software sells its Purify software for DCE which provides application checking (e.g., memory leaks, etc.). Each hardware vendor supplies its own development environment with support for DCE threads (e.g., SunSoft Workshop, HP SoftBench). HP also supports a distributed debugger which is very helpful, as well as an instrumented IDL/Runtime package for doing performance measurement.

5.5.1.2.3 Select preferred interface/tools

The DII COE should strongly consider the creation of a DCE encapsulation library as described in the CORBA migration strategy document.

Implementation Note: *A DCE interface library is being developed for the DII to simplify the development of DCE applications. Server routines will aid in server initialization and ACL management. Client routines will aid in client initialization and binding to a server, including starting a server on demand. DCE segment installation procedures and a sample application will also be provided for DII COE Version 3.0.*

5.5.1.3 Determine migration strategies for certain classes of applications

The primary goals are to integrate DCE security and other DCE services as needed.

5.5.1.3.1 Examine use of ORACLE with DCE

There are several approaches which can be employed depending upon the time frame. The first approach is to use a COTS product which intercepts client requests (by replacing the ODBC library on Windows machines) and uses a secure DCE server to access the database. This model is used by Open Horizon's CONNECT product and requires no recoding. The second approach is to use ORACLE's new release with integrated DCE. This provides Oracle clients with access to native DCE security facilities.

5.5.1.3.2 Integrating with DCE security

Existing applications can participate in the DCE security model by using a new set of DCE 1.1 APIs which allow clients and servers to communicate using Open Network Computing (ONC), but yet can exchange DCE credentials. The DII COE should develop a server library for handling the authorization/auditing functions in a uniform manner across all servers.

5.5.1.4 Determine if Ada bindings are required

The Army has developed Ada bindings for UNIXPROS DCE. These bindings are being evaluated and work is being done to convert this bindings to work under Transarc's DCE. A date for delivery of the TRANSARC Ada bindings has not been determined.

5.5.1.5 Determine security requirements for DCE applications

DCE offers many options in terms of security implementation. These options include what levels of security are used (i.e., integrity, confidentiality). Other issues relate to the names of server principals, attributes of Registry accounts, use of delegation, and auditing and authorization.

5.5.1.5.1 Determine the need for a standard or template ACL manager

Every secure service obtains the credentials of clients attempting to make use of its facilities. In DCE each server is responsible for making authorization decisions which is usually managed using DCE ACLs. The interpretation of permissions is assigned by each server and a standard algorithm is used to compare ACLs with credentials. It is important that The DII COE supply a standard interface for ACL management to ensure that servers are written securely. Managing the ACLs themselves is also part of this challenge. Choices include using the DCE 1.1 ACL

management which relies upon DCE backing storage management or using the DFS filesystem to store ACLs.

Implementation Note: *A standard ACL Manager and Reference Monitor are being provided for DII COE Version 3.0.*

5.5.1.6 Determine requirements for application auditing using DCE

DCE provides a non-centralized auditing service which can be used by any application server. Decisions which the DII COE must make are the selection of audit event numbers, inclusion in standard classes, default filter settings, what applications require auditing, and what types of operations must be audited.

5.5.1.7 Publish an application configuration strategy

Configuration of a DCE application involves its use of DCE facilities (e.g., directories required in CDS, security principals in the registry, files in DFS) and system facilities. One of the facilities provided by DCE 1.1 is the ability to remove configuration data from DCE applications. This data is kept in a database managed by the **dced** on each machine. Applications written to use these new APIs will be much more portable, easier to manage, and monitor.

Applications will follow the guidelines proposed for usage of CDS, Security, and DFS (mentioned elsewhere). Part of the DCE interface layer described in the CORBA migration strategy document will include functions for starting DCE servers on demand, monitoring their behavior, and managing their configuration. This can only be accomplished if applications use the same set of APIs for these purposes.

Configuration records are also accessible using DCE names. Each configuration database is appended to CDS under the subdirectory `/.../cellname/hosts/hostname/config/srvrconf`. Under this directory the same directories in CDS (namely applications/app-name) will be used to store application server configuration.

5.5.2 Development/Test

5.5.2.1 Test selected tools/interfaces

Several tools have been acquired and evaluated. To date only a limited set of tools will be provided as part of the DCE implementation. As more tools become available, they will be evaluated prior to fielding or segmentation.

5.5.2.2 Develop procedures for registering interfaces

Procedure registration occurs at two levels: registration within CDS and registration within the local endpoint mapper (**dced**). The standard DCE server registers as part of its startup logic. In DII, another mechanism will be used for services that are used only on demand (not needed at all times). These applications will register in CDS and in the **dced** configuration database as part of installation/configuration and will be started by the **dced** at the request of a client. When the server is started, it will complete the registration process by registering with the endpoint map. The benefit of this model is that host resources are conserved until needed (sockets, memory, CPU).

Implementation Note: *The procedures for registering interfaces was documented in Appendix X of the I&RTS, December 1995. These procedures will be supported by the standard interface library for DII COE 3.0.*

5.5.2.3 Develop guidelines for partitioning applications and defining interfaces

This task involves the development of a set of recommendations for creating distributed applications. The general steps are to define remote interfaces corresponding to remote operations. An example of a guideline is that operations should be selected that require some degree of processing at the server (to warrant the communications overhead).

Implementation Note: *Guidelines have been documented in Appendix X of the I&RTS, December 1995.*

5.5.2.4 Develop guidelines for application profiles and replication techniques

DCE services can be replicated for performance and availability reasons. Replicas can operate at the same site or at other sites, allowing system sites to automatically provide backup to each other. Profiles control the order of search for a service, as discussed in section 5.2.4.3. Guidelines should be developed describing efficient ways to implement and control replicated services.

5.5.2.5 Develop standard application security module and guidelines

The motivation for this effort was described earlier. Application developers will have to consider the number of individual objects that the server manages (i.e., protects by separate ACLs) and the granularity of operation expression (i.e., number of separately controlled operations). For example, a distributed filesystem has an ACL for every file being managed, whereas the DCE time services has only a single ACL for the time server. Applications should not create an

administrative nightmare.

Guidelines should include the following topics:

1) The use of a standard ACL manager or template ACL manager

2) Guidelines for use of the above. Once an ACL model has been selected, applications will have to create instructions for installation of default ACL controls.

3) Guidance on application trust of local and remote cell users/servers.

4) Guidelines for use of 'setuid' within servers to synchronize DCE and operating system identities. One of the issues that arises is when a DCE server needs to carry out operations on behalf of clients. The local operating system bases its decision on the effective identity of the invoker which is the server, not the originating client. If the server is considered a trusted application, then this can be dealt with by arranging for the server to run as a 'setuid process'. This privileged server can change its effective uid to that of the requesting client before performing the operation on its behalf.

5) Guidelines on use of secure RPC and levels of encryption/authentication used. GCCS operates using a secure private internet therefore it should suffice to use the standard DCE authentication level without integrity or privacy. Other DII systems will need to make their own security decisions based on their system architecture.

Implementation Note: *A standard ACL Manager and Reference Monitor are being developed for DII COE Version 3.0.*

5.5.2.6 Develop procedures for installing new DCE applications at sites

Procedures need to be developed that will allow a new DCE application to be installed at a site that is currently running DCE. The procedures involve "starting" an application and registering its server so DCE becomes "aware" of it. Procedures also need to be developed to de-install an application so all references are removed and a file/reference cleanup takes place.

5.5.2.7 Develop standard API and tools for auditing and audit reduction

DCE provides only the minimal tools for audit log reduction and there are no currently available audit-related products.

5.5.3 Integration

Includes actions required to prepare for delivery, including training.

5.5.3.1 Provide distributed application design input to the DII COE Integration Standard and COE Toolkit

Much of the guidance developed in this section belongs in the DII COE Integration Standard or Programmers Toolkit. The guidance will make use of lessons from prototype testing.

Implementation Note: *The DCE Appendix to the I&RTS, December 1995, contains initial guidance. The guidance will be revised as needed based on experience.*

5.5.3.2 Integrate applications with standard ACL manager

While the number of applications using DCE may be small for the implementation of DII COE v3.0, a standard ACL manager has been developed and will be available for the initial implementation of DCE. The implementation of a standard ACL manager will decrease development time and insure compatibility between new DCE applications and the COE.

5.5.4 Installation

The following activities must be performed at each development and operational site.

TBD

5.5.5 Operation

TBD

6.0 VERSION 4.0 PLANS AND ISSUES

DII COE Version 3.0 lays the foundation for full use of DCE in future versions of the DII COE. Following the applications guidance developed in accordance with this plan and published in the Integration and Runtime Specifications (IRTS) and Programmers reference manual, DII COE application developers can begin using the DCE services to develop truely distributed applications.

While DCE applications implemented in Version 3.0 will likely use standard DCE development tools, it is expected that more capable, easier-to-use tools will be available for Version 4.0. These could include OODCE or other object-oriented development tools. Ada95 bindings may also be available for Version 4.0.

Applications that currently use sockets or ONC RPC are the most obvious candidates to use DCE in Version 4.0. Another early focus should be on partitioning DII COE support applications to allow COE services on application servers to be accessed from any client platform. In some cases this will involve describing existing COE API's in DCE IDL; in other cases the existing API should remain undisturbed and the client/server interface should be inside the API wrapper -- transparent to the mission application.

Whenever possible, distributed applications should be designed to take advantage of the DCE capability for clients to transparently access alternative servers, allowing greater flexibility and reliability. Applications should also be designed to make proper use of the security mechanisms built into DCE.

Applications can also take advantage of the DFS as a mechanism for disseminating information, both across sites and within sites. The security and global authentication present in DCE DFS are also features that should be exploited in future DII COE releases. One advantage of using DFS is that it may not require software development or software releases. Increased use of DFS can occur at any time and does not have to wait until Version 4.0.

APPENDIX A - Acronyms

ACL	Access Control List
API	Application Programming Interface
C2	Command and Control
CDS	Cell Directory Service
CFS	Center for Standards
COE	Common Operating Environment
COTS	Commercial off-the-shelf
DCE	Distributed Computing Environment
DCWG	Distributed Computing Working Group
DFS	Distributed File System
DISA	Defense Information Systems Agency
DNS	Domain Name Service
DTS	Distributed Time Service
FTP	File Transfer Protocol
GCCS	Global Command and Control System
IDL	Interface Definition Language
IOC	Initial Operating Capability
IT	Information Technology
ITS	Information Technology Standards
JIEO	Joint Interoperability and Engineering Organization
NCA	National Command Authority
NFS	Network File System
NIC	Network Information Center
ONC	Open Network Computing
OODCE	Object Oriented DCE
OSF	Open Software Foundation
RPC	Remote Procedure Call

APPENDIX B - DCE Cell Structure

B1.0 Background

A **Cell** is the basic unit of operation and administration in DCE. As in nature, the cell is the basic building block of the DCE structure. A cell is a group of users, systems and resources that typically have a common purpose and share common DCE services. In DII, they should share a logical connection, functional relationship, or group association. Since GCCS will be the first system using the DII COE and DCE, the cell structure will be built upon the CINC/Service administrative infrastructure. Other DII systems, like GCSS, will have to develop cell structures that fit their administrative and operational structures.

Generally the first question to consider is whether DII should be one cell or many cells? There are many divergent views on the proper size of a cell. Could DII be composed of one large cell? There are many tradeoffs that must be evaluated when you look at a mono-cell organization or a multi-cell organization. These tradeoffs include cost, operational complexity, and service levels. More importantly, the DII is composed of systems and organizations that have complimentary missions but do not have a need for resource sharing. Due to the size of the DII and its complexity, a one cell structure is not feasible.

But what criteria should be used for determining the size and number of the cells that will make up the DII? The key is resource sharing and shared DCE services. The DII systems like GCCS and GCSS will have an inherent need for resource sharing and for inter-organizational communications. The interaction between sites/organizations will also have a significant impact on the size and number of cells.

B2.0 Cell Tradeoffs

As noted, the basic decision of how many cells should be created depends upon the size of the system and the complexity. Small systems or organizations should consider adopting a single cell structure. A one cell DCE structure for any system would have a central site responsible for security and cell administration. This centralized structure would have performance implications since all of the other sites would have to perform all updates against the centralized security and administration services. By replicating directories into distributed clearinghouses and by distributing read-only versions of the security database, you can overcome some of the performance problems of the one cell structure. The advantages of a one cell configuration are lower costs, less complex structure (management), and more efficient communications between members of the cell. If all of the sites were in one cell, they could more easily communicate and share resources because as members of a cell they use the same security and administrative servers.

There are disadvantages to a one cell configuration for GCCS. The biggest disadvantage is that a catastrophic failure of any application or service will affect all of the members of the cell. This is particularly true of the security and directory services since only one master (read/write) copy is maintained. While you can replicate these directories, the duplicates are read only copies. In addition, the configuration management and security administration of a single large cell would be a very difficult and time consuming task. Security administration for a single cell configuration could be much too large to be handled efficiently.

A multi-cell configuration has distinct advantages over a mono-cell configuration. A multi-cell configuration provides the capability to place applications in separate cells to isolate them from individual cell failures and lets you customize the DCE services to individual groups. In a multi-cell environment, cells can run different versions of the same software because they are separate implementations and not centrally controlled. In a multi-cell configuration, replicating DCE services can improve performance and reliability of your applications but it can raise the cost and complexity of managing the DCE environment.

Some of the disadvantages of the multi-cell configuration are cost, complexity, and intercell communications. Multi-cell configurations also have the disadvantage of intercell communication and authentication. For cells to communicate and share resources, a trust relationship needs to be established between each of the cells. The greater the number of cells, the greater the number of trust relationships that must be established. For four cells to have mutual trust between all members of the system, sixteen trusted relationships would have to be formed. This increases exponentially with the increase in the number of cells.

B3.0 Transitive Trust and Hierarchial Cells

The problem of mutual trust and authentication becomes a huge administrative burden as the number of cells increases. DCE 1.1 was designed to solve this problem and reduce the burden significantly. In DCE 1.1, you could organize cells in a hierarchial manner somewhat akin to the way an organization is laid out in a hierarchial manner. For example, a company has a head office that has different departments reporting to it. The departments have sections that it administers.

What was supposed to happen in DCE 1.1 one was that if you had a hierarchial structure, you could implement transitive trust. Transitive trust is the mechanism that allows already established trust relationships to be passed on to higher levels in the hierarchy. For example, if the accounting department has two sections organized underneath it with which it shares resources. It has a trusted relationship with its sections. If the manufacturing department has the same relationship with its sections, it can establish a trust relationship with the accounting department such that all of the sections of the accounting department now trust all of the sections in the manufacturing department. Because the two departments trust each other and the departments

trust their sections, then that trust transits both departments and all sections.

Hierarchial Cells and Transitive Trust are required if a large number of cells are going to be implemented that need to have intercell communication and authentication. Without these capabilities, the administrative burden becomes unmanageable. Unfortunately, these mechanisms were not fully implemented in DCE 1.1 and are only partially effective. It appears that these mechanisms are not high on the priority list to get implemented in the near future.

Regardless of when these mechanisms are fully implemented, new DCE implementations should be designed around hierarchial cells. This is necessary to provide the capability to use Transitive Trust in the future. The expansion of any system will depend on the ability to add new cells and establish trust relationships. The administrative burden associated with a large multi-cell structures is too great without Transitive Trust.

B4.0 Summary

The design of a cell structure for any system is a balancing act between efficiency and performance. The goal is to have the greatest efficiency with best performance. Generally, in the macro view, this means having the minimum number of cells possible because each new cell will increase the administrative burden for intercell communication and authentication. Hierarchial cells and transitive alleviate this problem for the most part but these mechanisms have not been fully implemented in the current releases of DCE.

APPENDIX C - Proposed Tools Analysis

C1.0 Cell Manager

Chisholm offers a management product called the Cell Manager. It is available for all DCE platforms and provides a graphical administrative interface.

Its benefits include the ability to do remote management even when DCE 1.0 limitations precluded this capability (using remote agent technology). It also can perform initial client configuration for all DCE systems.

Its shortcomings include its lack of extensibility, it doesn't currently support 1.1, its use of remote agent technology, and its lack of support for more than one cell.

C2.0 Cell Factory

UNIXPROS under contract with the ARMY has developed a technology called Cell Factory which is used to install and configure DCE cells rapidly and dynamically.

C3.0 NASA Tools

NASA Langley Research Labs has developed a small suite of public domain tools which act as administrative aids. The following is an excert of the tools developed by NASA Langley Research Center as part of their ICE (Intetrated Computing Environment) project.

DCE XDM Version 1.0 (based on X11R5-pl26) (LSS-1995-0010)

This is the source for a DCE authenticating version of the X11R5 (pl26) xdm. Not much to say here, really. It's a drop in replacement for the distributed xdm binary. It ignores your local password file, and assume that anyone in the DCE registry can log into your machine if they can provide a valid password. This may or may not be what you want.

Tacl is a graphical interface to DCE's acl_edit utility. It is written using tkperl, and will run on any system that has tkperl and DCE installed. Although designed primarily for use with DFS ACLs, it will work properly on any DCE ACL so long as acl_edit works properly.

DCEPerl-0.5 is a beta release of a Perl5 package for calling DCE registry functions. You'll see from the comments in the DCE.xs file which functions we have not needed yet (they are the ones

I have marked as still necessary to write), but we have found what is there to be adequate for most of our day-to-day user accounting functions.

Passwd_export is a perl script that utilizes DCE.pl (a locally developed DCE perl module) to export the contents of the DCE registry into UNIX password and group files. This version is easily extensible to support local political decisions and arbitrary registry structures, and offers greater expandablity than the vendor supplied version.

DCE**passwd** is a short C program that is intended as a wrapper for the vendor supplied **passwd** command; it attempts to keep the DCE registry in sync with the local **passwd** file. There is a compile option to make it only update the DCE registry, which simply leaves you with a **passwd** command with the standard UNIX interface (the user would otherwise have to make the change to their password).

DCErsh is a version of the popular rsh/rshd remote shell execution programs which conveys the senders DCE credentials to the server. This allows the server to execute on behalf of the client using its DCE identity.

An NIS to DCE gateway is currently under development.

C4.0 SCO's Distributed Administration Service

The Distributed Adminstration Service also offers a comprehensive graphical environment for management of DCE Cells. Its primary benefit is that it was built using the DCE version 1.1 control program (**dcecp**) and is the only product that was designed to be highly extensible both functionally and graphically. The fact that it is produced by SCO does not restrict its applicability to DCE management. In all likelihood, it won't require an SCO machine for operation.

Its shortcomings include its lack of DCE version 1.1 functions, and that SCO itself has nearly dropped support for this product. Another company is currently negotiating with SCO for product rights and would carry forward with the version 1.1 functions and other enhancements including broader platform support.

APPENDIX D - Proposed Distributed File System Utilization

D1.0 Overview

This appendix describes DCE Distributed File System (DFS) and proposes a way to implement DFS within GCCS. Recommendations that are specific to GCCS are shown in italics. DCE DFS is a distributed client/server application that presents the user on any GCCS system with a global view of a set of files and directories (a file system), independent of machine boundaries. This global view is called the DFS filespace. Experience gained from the GCCS DFS implementation will be used to develop a DFS implementation plan for the DII.

DFS is considered distributed because files can be physically stored on many different machines, in various geographical locations and potentially in different cells, yet still be available to users on every machine. DFS allows users to share files stored on computers in a network as easily as files stored on a local machine. Regardless of where files are stored, users perceive a single filespace. Access to this filespace is protected using DCE security credentials and a sophisticated ACL mechanism.

GCCS will deploy DFS to make accessable a shared secure filesystem across all GCCS sites. The usefullness of this capability will be discussed later in this appendix.

DFS is supported by a set of DFS clients and servers. DFS Servers house the physical file system while DFS clients access these servers. DFS client software is resident in the operating system of a client machine which then provides transparent access when presented with a filename of the form `/.../cellname/fs/filename`. Using these pathnames, it is possible to access the file services of any DFS server regardless of its cell.

In GCCS , only a single cell will contain DFS servers, (`/.../gccs.mil`). This cell will be managed by the Joint Staff (or DISA on their behalf) and will have a server placed in one site in each theatre (DISA, TRANSCOM, EUCOM, and PACOM). This server will be physically located at a theater CINC site, but will not be part of the theatre's cell, and will be jointly managed by local administrators and the Joint Staff (see discussion on administration below). Using DFS in this fashion allows GCCS to take advantage of DFS's replication capability while limiting the impact of DFS administration to a small number of individuals. This in no way precludes global GCCS client access. Figure D-1 summarizes the GCCS DFS implementation.

The hardware configuration for this dedicated file server machine is identical to the Map Server. This machine will also serve as the repository for a global CDS space.

The recommended approach includes the following:

- Implement limited DFS infrastructure in Version 3.0; one server per theater.
- Use DFS for sharing files between sites.
- Require new applications that would use NFS for inter-site transfers to use DFS instead.
- Require existing applications that already use NFS to move to DFS by release 5.0.
- Move some UNIX files into DFS, concentrating on files that are shared or require additional security.
- DO NOT move all UNIX files to DFS in Version 3.0.
- Allow mobile users to move their home directories to DFS.
- Use DFS to distribute Application Segment Binaries/Libraries to sites.
- Allow sites/systems to install additional filesets into DFS as required.

Figure D-1. Summary of DFS Implementation

D2.0 DFS Server Machines

DFS server machines run processes that provide a range of services, such as making data available and monitoring and controlling other processes. These server machines are categorized by the processes they run (the roles they assume). For example, a server machine that runs the processes necessary for storing and exporting data assumes the role of a File Server machine. The processes of a File Server machine include the Fileset Server (which provides an interface to the DFS commands and components used to manipulate filesets), and the File Exporter (which runs in a modified kernel to make DFS files available to the global namespace).

Other server machine roles include the following:

- A System Control machine that updates other server machines with identical versions of system configuration files;

- Binary Distribution machines that distribute system binaries to other machines with the same CPU/operating system type (*GCCS will not require this role because the operating system binaries will not change very frequently*);

- Fileset Database machines that house the master and replica versions of the Fileset Location Database (FLDB) where information about the location of system and user files is maintained; and

- Backup Database machines that house the master and replica versions of the Backup Database where information used to back up and restore system and user files resides.

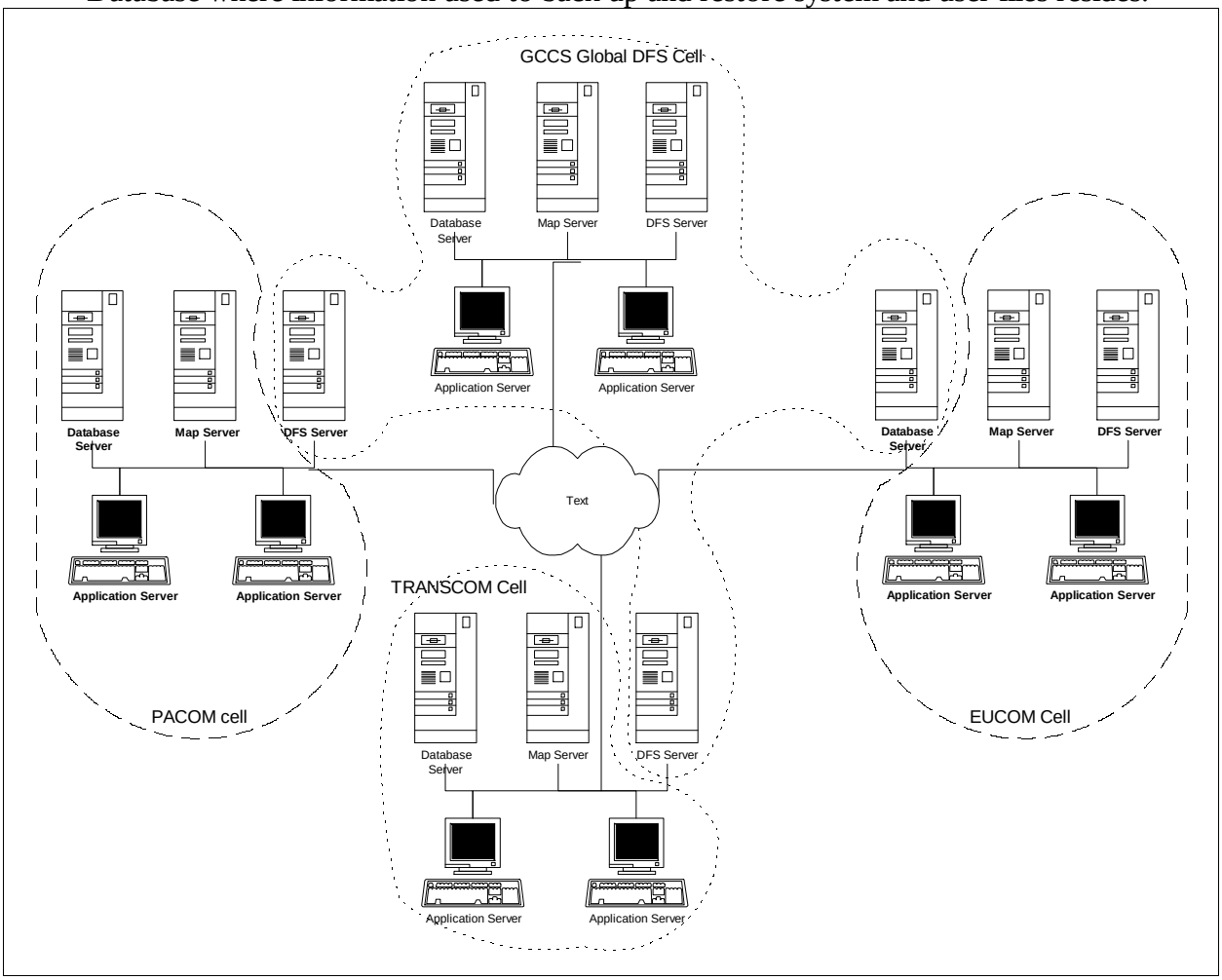

Figure D-2. GCCS Global Cell

The GCCS DFS Servers will be located in a single cell. The DISA cell will house the writeable root of the DFS space. In addition, the DISA cell and the server in TRANSCOM will maintain the central DFS internal databases, and will therefore serve as Fileset DB, and Backup DB. All server machines will act as file servers. The System Control role is not required where each server defines its own administrative domain (see below). Appendix E on DCE deployment depicts these roles and process names in tabular form.

D3.0 DFS Client Machines

Client machines use a modified kernel that maintains contact with the file server processes running on server machines. This collection of kernel modifications on a client machine is known as the Cache Manager. The main duty of the Cache Manager is to translate file requests made by application programs on a client machine into Remote Procedure Calls (RPCs) to File Exporter processes on File Server machines.

When the Cache Manager receives requested data from a File Exporter, it caches the data (stores it on disk or in memory) before passing it to the application program that requested it. In addition, DFS ensures that the Cache Manager always has access to the most current copy of the data. If the central copy of the file containing the data changes, the Cache Manager retrieves the newer version of the file the next time data is requested from the file (or in the case of read-only data, within a configurable period of time). The user does not have to direct the Cache Manager to keep a current copy; the Cache Manager's actions are automatic and completely transparent to the user.

The Transarc DCE base services includes the DFS client software. Since the DCE client software is being segment into the DII COE kernel, every DII COE client will have the capability to be a DFS client.

Instead of providing direct access to DFS, GCCS will provide each site with software acting as a Secure NFS-to-DFS Translator. This will allow client machines to use NFS to obtain DFS information without compromising the DCE cell-wide security model. NFS is much lighter, much better understood, and is available from many suppliers for any platform.

The NFS to DFS Gateway provides Network File System (NFS) client users authenticated access to the Distributed File Service (DFS) filespace in the Distributed Computing Environment (DCE). The ability of NFS clients to access the DFS filespace from a DFS client is already inherent in DFS. However, without the NFS to DFS gateway, NFS client users are limited to unauthenticated access because the authentication information for DFS and NFS is different. The NFS to DFS Gateway provides a bridge between the diverse authentication information and equates an NFS client user with an authenticated DCE principal. This gives NFS client users authenticated access to the DFS filespace and to the features of DFS.

To provide file access, the NFS client mounts the exported DFS filespace from the gateway machine (e.g., `mount gateway:/.../gccs.mil/fs /gccs`). In DCE, end users are identified by principal names. In NFS, client requests are identified by an IP address and USERID pair. Because the NFS to DFS Gateway resides on a separate machine, NFS users must login to this translation point, and issue the **dfsiauth** command. The **dfsiauth** command describes the address of the NFS client user's machine and the user's id (uid) on that machine, and authenticates with DCE, declaring a DCE principal and entering the password. When the NFS to DFS

Gateway receives an NFS request, it includes the NFS client user's machine address and uid of the request. The NFS to DFS Gateway embeds the DCE credential information in the request, enabling DCE authentication.

NFS client users may use the NFS to DFS Gateway without making changes to their existing software if they use the **dfsiauth** command on the NFS file server. In addition, an API library is provided with the NFS to DFS Gateway so client/server applications may be developed that establish authentication mappings from the native NFS client to the NFS to DFS Gateway system. Such applications would eliminate the need for users to login to the NFS to DFS Gateway machine to establish their mappings. *EM would be a good candidate to provide this integration.*

One machine in each GCCS site, configured as the NFS to DFS Gateway, would be the only machine requiring DFS Client capability. The DFS client requires no special administration with the possible exception of initial tuning of its Cache Manager parameters. This machine is acting as a client on behalf of every machine in the site. If having a single gateway proves to be a performance or reliability problem, the decision to rely upon a single gateway can be re-visited. Options include using multiple gateways or native DFS client access.

Each file server machine will also act as a client machine in order to provide visibility into its DFS files.

D4.0 DFS Administration

D4.1 DFS Administrative Domains

DFS further extends the concept of a DCE cell by providing DFS *administrative domains*. An administrative domain is a collection of associated server machines from the same cell configured for administration as a single unit. A cell can include a large number of machines; administrative domains provide a means of simplifying the administration of many DFS machines in a single DCE cell by organizing a subset of the cell's machines into smaller administrative units which are usually administrated by different sets of individuals. In addition to simplifying the management of DFS in a DCE cell, administrative domains bring fine levels of granularity and flexibility to DFS administration in general. A cell can have one or more administrative. *Because each theater site will have some responsibility for administration of their machines, each theater file server will be placed in a separate domain.*

D4.2 DFS Administrative Lists and Groups

Administrative lists are files used with DFS administrative domains to determine which individuals are allowed to issue commands that affect specific processes and data. Being a member of an

administrative list is analogous to having the permissions necessary to issue requests to the associated server process. Individual users can be placed on administrative lists to grant them the administrative privileges associated with the lists. Groups of users can also be placed on administrative lists to grant the privileges associated with that list to all of the members of the group simultaneously; i.e. the members of a group have all the privileges associated with any administrative lists in which the group is included. *To limit the amount of training required, administrative lists in GCCS will be limited to administrative personnel at the theater file server sites.*

In GCCS, the DISA cell containing the DFS servers will employ an administrative domain for each of the servers. The domain lists will be created such that each theater can manage its own server, and DISA personnel will also have management access but the site administrator will not have management authority over the DISA servers. More specifically, each site can add its own filesets and perform load balancing, tuning, and backup, but the filesets which are propagated from the DISA will only be modifiable by DISA staff. For example, policy documents, procedures, and application segments will disseminated using DFS's replicated fileset capability. These filesets are part of the DISA domain and can only be added or changed by DISA personnel.

D5.0 DCE Local File System

Files on a DFS server are stored in the DCE Local File System (LFS). DCE Local File System (DCE LFS) is a high-performance, log-based file system. DCE LFS supports the use of aggregates. A DCE LFS aggregate is physically equivalent to a standard UNIX disk partition, but it also contains specialized metadata about the structure and location of information about the aggregate. DCE LFS aggregates also support the use of filesets, which is a hierarchical grouping of files managed as a single unit. DCE LFS filesets can vary in size but are almost always smaller than a disk partition. In DCE LFS, multiple filesets can be stored on a single aggregate, providing flexible disk usage. GCCS will not use or support non-LFS partitions due to the limitations in managing these partitions.

With DCE LFS, the potentially small size of filesets allows them to be easily managed for maximum system efficiency. Also, each DCE LFS aggregate can store multiple DCE LFS filesets. A system administrator can move filesets either from one DCE LFS aggregate to another or from one machine to another for load-balancing across machines. If the complete contents of a user's home directory are stored in one fileset, the entire directory moves when the fileset is moved. Although there will be limited use for these capabilities in GCCS, they will be useful as DFS usage grows in the future.

Each DCE LFS fileset corresponds logically to a directory tree in the file system. Each fileset maintains, on a single DCE LFS aggregate, all of the data that comprises the files in the directory

tree. For example, if you maintain a separate fileset for each user's home directory, you can keep a user's files together but separate from those of other users.

The place at which a DCE LFS fileset is attached to the global filespace is called a mount point. A mount point looks and acts like the root directory of the fileset. This correspondence between a directory and fileset also simplifies the process of file location. A mount point identifies a fileset by name so that DFS can automatically locate the fileset, even if the fileset is moved between aggregates or machines. Mount points also control how a DFS client reaches filesets. This is especially relevant for replicated filesets, discussed below.

Each DCE LFS fileset has a fileset quota associated with it. A fileset's quota specifies the maximum amount of disk space the information in the fileset can occupy. Quota is set on a per-fileset basis, so it can be increased for filesets that contain more data and decreased for filesets that do not need the additional disk space.

D6.0 DFS Replication

DCE LFS allows DCE LFS filesets to be transparently replicated (copied). When a DCE LFS fileset is replicated, read-only copies of it are placed on multiple server machines. The unavailability of a single server machine housing a replicated fileset does not usually interrupt work involving that fileset because copies of the fileset are still available from other machines. The replication of commonly used reference, configuration and binary files on multiple server machines greatly reduces the chances of their being unavailable because of server machine outages. Replication also prevents a machine from becoming overburdened with requests for files from a frequently accessed DCE LFS fileset. Replication is supported only for DCE LFS filesets, not for non-LFS filesets (file systems on non-LFS partitions). *For GCCS, most DFS filesets will be replicated on all DFS servers. This provides in-theater access to all DFS files, while allowing access from all sites even if the regional file server is down.*

In DCE LFS, there is exactly one writeable fileset and potential many read-only copies. *The DISA site will contain a set of writeable filesets some of which will be replicated in each theatre. These filesets will be used to distribute configuration, applications and other GCCS information to each of the geographic areas. Each of the theatres will replicate each of these filesets.*

Each of the theaters will not use replication locally as theatre's are assumed to contain a single DFS server in 3.0. However, each site will maintain a set of writeable filesets (one per site in the theatre) which will be used to maintain the backup DCE configuration and DCE internal databases. These filesets will be replicated read-only back to DISA where backups, compliance, and troubleshooting can be performed.

Two types of replication are available with the DCE LFS: Release Replication and Scheduled Replication. With Release Replication, an administrator can issue a command to copy a source fileset to the server machines housing its read-only replicas on demand to reflect the current contents of the read/write fileset. This type of replication is useful either if the fileset seldom changes or if there is a need to closely monitor the replication process.

With Scheduled Replication, an administrator can specify replication parameters that dictate how often DFS is to automatically update replicated filesets with new versions of source filesets. This type of replication is useful if it is preferred to automate the process and there is no need to track exactly when releases are made. Both types of replication produce the same result : source filesets are copied to different server machines. The system administrator chooses which type of replication to use with each fileset.

GCCS filesets which represent relatively static information will use release replication as they change infrequently and there is a need to monitor when a change is propagated. Filesets created by local site to contain their site's configuration and backup data can use scheduled replication on an infrequent basis. Although scheduled replication can be set to replicate if needed every second, this is not the intention of this mechanism.

In general, replicate only those DCE LFS filesets that meet the following criteria:

- The files in the fileset are read much more often than they are modified.

- The files in the fileset are heavily used. For example, binary files for text editors or other popular application programs. Replicating the fileset lets you distribute the load for the files that it contains across several machines.

- The files in the fileset must remain available. By replicating the fileset on multiple File Server machines, even if one of the machines that houses a replica of the fileset becomes unavailable, replicas are usually still available from other machines.

- The fileset is mounted at a high level in the cell's file tree; for example, `ROOT.DFS` and its subdirectories.).

A set of configuration parameters for filesets and cache managers will need to be developed for replicated fileset management.

D7.0 DFS Backup System

DFS provides two methods of managing backups: the DFS Backup System and backup filesets.

With the DFS Backup System, an administrator can copy data from filesets to tape and restore the data from tape in the event that the data is lost. Information about backups and tapes is maintained in the replicated Backup Database. The database itself can be copied to tape and restored in the event of its corruption. Backups of both DCE LFS filesets and non-LFS filesets are supported.

An administrator can perform both full and incremental backups, or dumps. A full backup copies all of the data in a fileset to tape; an incremental backup copies only those files that have changed since the last full backup to tape. A backup schedule, or dump hierarchy, records the specified filesets to be included in a backup.

The DISA site will perform regular media backups of all writeable filesets and read-only replicas (for sites which desire this service or for sites without these capabilities).

Backup filesets capture the state of source data at the time the backup is made; they do not involve the Backup System. An administrator can create a backup version of a user's DCE LFS fileset and mount it as a subdirectory of the user's home directory, naming it something appropriate such as `.OLDFILES` or `.BACKUP`. The user can then, without assistance, restore to a read/write fileset any files deleted or changed since the backup fileset was made. Users cannot change the data in their backup filesets, but they can copy the data to a regular directory in a working, read/write fileset and use it there.

Site administrators will use this backup capability to maintain the previous DCE data files and configuration backup. Unlike performing a copy, a backup fileset requires a fraction of the storage space (only the blocks that have actually been changed).

D8.0 DFS Naming

D8.1 Choosing Fileset Names

Each top-level directory in `/.../cellname/fs` usually corresponds to a separate, mounted fileset (mounted filesets can also occur elsewhere in the file tree). Subdirectories of `/.../cellname/fs/directory_name` can be either standard directories or mount points to separate filesets. For simplified administration, group the directories and their contents into small, easily managed filesets.

Each fileset has a name unique to the cell in which it resides. Fileset names are stored in the FLDB (replicated fileset location database). A fileset's name is not necessarily the same as the name of its mount point, although you can assign the same name to a fileset and its mount point. Fileset names are only the concern of administrators because filesets are the basic unit of

administration (similar to a file system in UNIX).

There is a 111-character limit on the length of fileset names. However, because a 9-character `.READONLY` extension is added when you replicate a fileset, you need to specify fileset names that contain no more than 102 characters. When creating filesets, do not add the `.READONLY` and `.BACKUP` extensions yourself; DFS automatically adds the appropriate extension when it creates a read-only or backup fileset. (DFS reserves these extensions for use with read-only and backup filesets, so you cannot create a fileset whose name ends with either of these extensions.)

For simplified administration, however, a fileset's name needs to do the following.

- Reflect the fileset's contents
- Reflect the name of the fileset's mount point
- Be consistent with other filesets that contain similar types of data so that you can easily manipulate groups of filesets when using the DFS Backup System
- *For GCCS, each fileset name should be prefixed with the site name if it is site-related, or with a theatre name if theatre related. Filesets that are GCCS global should not be prefixed.* For a replicated fileset, the name is taken from the read-write copy.

It is helpful to use a common name component for related filesets. The following list summarizes this type of naming scheme:

- Use the COMMON.type component for common filesets. For example, use COMMON.ETC for common configuration files (mounted at `/.../cellname/FS/COMMON/ETC`), and `COMMON.FORMS` for common forms (mounted at `/.../cellname/FS/COMMON/FORMS`).
- Use the `USER`.username component for all user filesets. For example, use `DISA.USER.TERRY` for user `TERRY`'s fileset (mounted at `/.../cellname/FS/DISA/USR/TERRY`).
- Use the `PUBLIC.username` component for each user's public fileset. For example, use `DISA.PUBLIC.TERRY` for `TERRY`'s public fileset, which contains information the user wants to make available to everyone. The `PUBLIC.TERRY` fileset is mounted at `/.../cellname/FS/PUBLIC/DISA/TERRY`.
- Use the `sys_type.distribution_dir` component for operating system-specific

filesets. For example, use SPARC_SOLAR23 for Solaris 2.3 binary files (mounted at /.../cellname/FS/APP_SEGMENTS/JOPES/SPARC_SOLAR23/BIN, and SPARC_SOLAR23.LIB for Solaris 2.3 libraries (mounted at /.../cellname/FS/APP_SEGMENTS/JOPES/SPARC_SOLAR23/LIB. If DFS eventually gets used as the file system housing the applications rather than the distribution method, symbolic links can be created from the /BIN and /LIB directories (or their equivalents) on the local disk of a workstation to these DFS mount points or the DFS directories can be added to the PATH variable.

D8.2 Selecting directories

Closely related to selection of file set names is the selection of mount points (i.e. how the directories are used).

In GCCS, because all sites are sharing the same global file space, it is recommended that GCCS create upper level directories which mimic the hierarchy in place for cells for any data which is specific to a theatre or site. For example, a top level directory will be created for each of the CINCS below which are directories for Army, Navy and Marine as well as directories for each of the departments.

- *Directories relating to GCCS applications should be placed under /.../gccs.mil/app_segments/APPNAME with each application segment being resident in a fileset under this directory.*

- *Another top level directory (e.g., /.../gccs.mil/gccs) should be used to house the configuration, data files, installation procedures, documents, problem logs, configuration changes, etc. which are global to all gccs or private to each site.*

- *Other top level directories should include project directories.*

D8.3 Setting Up User Filesets

Each user has a unique DCE account. *It is possible in GCCS to create a single, separate fileset for each user and mount the fileset at /.../cellname/FS/SITE/USR/username*, where username is the name of the user who owns the fileset. For example, assign the name USER.TERRY to the fileset for the user named TERRY and mount the fileset at /.../cellname/FS/DISA/USR/TERRY. The user's home directory contains all of the files, subdirectories, and mount points in the fileset named DISA.USER.TERRY.

This would allow any GCCS user to maintain information of any kind, (e.g., phone list, e-mail,

assignments, ...) in DFS and to access it from any machine within GCCS. While GCCS does not require a user to use this fileset, it wastes no resources. *For users who travel between sites, being able to access their mailboxes or folders securely and transparently from any GCCS site without having to do a remote login is very useful.*

D9.0 System Administration -- A Task Overview

The administration of DFS can be divided into the following general types of tasks:

• FILESET MANAGEMENT -- Efficiently creating, deleting, and organizing the filesets in a cell and performing appropriate backup and restores. Maintaining fileset replicas, performing replication, monitoring growth of filesets and allocation of aggregate storage, and fileset quota management.

• SYSTEM MANAGEMENT AND CONFIGURATION -- Monitoring the performance of the file system software and making adjustments as necessary. DFS provides a utility SCOUT to assist in determining the behavior of the DFS service.

• SECURITY ISSUES -- Establishing the correct procedures and policies to ensure the security of the file system. This requires setting initial ACLs, and performing a periodic ACL audit. The security mapping between machine, USERID and DCE account also needs management for gateway machines.

• FILE DATA MANAGEMENT -- Managing the content of files which are accessible throughout GCCS.

• ACCREDITATION -- Performing analysis of information and configuration maintained about each site and stored in DFS to determine its accreditation.

There are three groups of administrators responsible for DFS management. *System and security administrators at the DISA site, administrators and each of the other DFS server sites and administrators at other sites who own filesets (e.g., personnel maintaining the SORTS database files). For GCCS, local site administrators will have no DFS administration responsibilities. However, the local site security administrators will be responsible for any security administration of DFS files.*

DISA administrators are responsible for all writeable filesets located at the DISA, and the file server machine housed at their site. They would perform all of the tasks listed above for filesets and equipment under their jurisdiction. DISA staff would also perform ongoing accreditation checks.

Each of the DFS server sites maintain responsibilities for writeable filesets maintained at their site. These filesets include filesets they are sponsoring for all sites in their theatres. Responsibilities include fileset management, system management, security issues, and limited file data management.

Each site's administrators is responsible for file data management, and security management of any writeable fileset being sponsored by the theater. Each site must perform basic system management and backup of their gateway machine.

APPENDIX E - Proposed DCE Server Allocation

The following proposal is based on a standard DCE configuration for both a small site and a large cell. In this example, the terms site and cell are used interchangably.

Small sites all have a Data Server and 2 App. Servers. The Data Server be the DCE Server housing the Security, CDS, DTS server and NFS to DFS gateway. The rationale is that this server will be up all the time, will be heavily guarded (it won't go anywhere) and has the resources to support DCE. Each of the App. Servers will run a DTS server as well, which gives us our minimal DCE DTS requirements.

Larger sites will have an additional Data Server and more App. Servers. Additional Data Servers will house Security Replica Servers and CDS Replicas and DTS Servers. The rationale is that for a larger site, its probably worth having the additional failsafes provided by replication. Additional App. Servers will also have DTS servers until there are 5 on a single LAN. After that point, no additional DTS servers are required.

Table E-1. Allocation of DCE Servers in Site Cells

	dced	secd	cdsd	dtsd	gdad	DFS Gateway	
Data Server	Y	Master	Initial	Server	Yes (1/theatre)	Y	
Data Server(N)	Y	Replica	Replica	Server	No	N	
Appl. Servers(2)	Y	N	N	Server	No	N	
Appl. Server(N)	Y	N	N	Server <=5	No	N	
Other	Y	N	N	Clerk	No	N	

All other machines should be running the DCE core. 486 workstations or Macintosh computers can act as DCE clients or use a DCE gateway product. There are COTS products available to meet these needs.

APPENDIX F - DCE Implementation for GCCS

F1.0 Overview

The DII guidance provided in this document generally applies to the implementation of DCE for GCCS. This appendix will address any GCCS specific items that are not addressed in the body of this document or where the general guidance can be replaced with GCCS specifics.

F2.0 DCE Cell Structure

Generally the first question to consider is whether GCCS should be one cell or many cells? There are many divergent views on the proper size of a cell. Could GCCS be composed of one large cell? There are many tradeoffs that must be evaluated when you look at a mono-cell organization or a multi-cell organization. These tradeoffs include cost, operational complexity, and service levels.

If the sole purpose for establishing a cell is for resource sharing and shared DCE services then a one cell organization makes sense. All of the GCCS Initial Operating Capability (IOC) sites share the same basic functions (applications) and have a singular purpose: to support Joint Command and Control (C2) of US Forces. A one cell DCE structure for GCCS would have a central site responsible for security and cell administration. This centralized structure would have performance implications since all of the IOC sites would have to perform all updates against the centralized security and administration services. By replicating directories into distributed clearinghouses and by distributing read-only versions of the security database, you can overcome some of the performance problems of the one cell structure. The advantages of a one cell configuration are lower costs, less complex structure (management), and more efficient communications between members of the cell. If all of the IOC sites were in one cell, they could more easily communicate and share resources because as members of a cell they use the same security and administrative servers.

There are disadvantages to a one cell configuration for GCCS. The biggest disadvantage is that a catastrophic failure of any application or service will affect all of the members of the cell (GCCS). This is particularly true of the security and directory services since only one master (read/write) copy is maintained. While you can replicate these directories, the duplicates are read only copies. In addition, the configuration management and security administration of a single large cell would be a very difficult and time consuming task. Security administration for a single cell GCCS configuration would be much too large to be handled efficiently.

A multi-cell configuration for GCCS has distinct advantages over a mono-cell configuration. A multi-cell configuration provides the capability to place applications in separate cells to isolate

them from individual cell failures and lets you customize the DCE services to individual groups. In a multi-cell environment, cells can run different versions of the same software because they are separate implementations and not centrally controlled. In a multi-cell configuration, replicating DCE services can improve performance and reliability of your applications but it can raise the cost and complexity of managing the DCE environment. Some of the disadvantages of the multi-cell configuration are cost, complexity, and intercell communications.

The basic reason for not implementing GCCS in a single cell is that a catastrophic failure in any of the DCE services will affect the entire cell (all members). While you can protect against this problem by replicating services within a cell, it is still possible that an entire service can have a catastrophic failure. In addition, the organizations that compose the GCCS community have an established hierarchial structure and business process that does not support a one cell configuration. For these reasons, the GCCS DCE should not be implemented as a single cell but should have multiple cells.

F3.0 Proposed GCCS Cell Structure

How many cells? The GCCS Initial Operating Capability (IOC) sites are listed in figure B-1. These sites are composed of different organizational entities within DOD: CINCs, Components, Services, and Agencies. A cell usually consists of nodes that have a common geographic reference (i.e., on the same LAN or WAN). But geography is not an overriding consideration for membership in a cell or where cell boundaries are established. The boundaries of a cell can be determined by a multitude of factors but the basic considerations are: purpose, administration, security, and overhead.

ACRONYM	ORGANIZATION	ACRONYM	ORGANIZATION
ACC	Air Combat Command	MARFORLANT	USMC Forces,Atlantic
ACOM	Atlantic Command	MARFORPAC	USMC Forces-Pacific
AFMC	Air Force Materiel Cmd	MSC	Military Sealift Cmd
AMC	Air Mobility Cmd	MTMC	Military Traffic Mgt Cmd
ANMCC	Alt. Nat'l Mil. Cmd Ctr	NAVCENT	Navy Central Command
ARCENT	Army Central Cmd	NAVEUR	US Navy-Europe
AREUR	US Army, Europe	NMCC	National Mil Cmd Ctr
ARPAC	US Army, Pacific	PACAF	Pacific Air Forces
CENTAF	Air Force Central Cmd	PACOM	Pacific Command
CENTCOM	Central Command	SOCOM	Special Operations Cmd
CINCLANTFLT	CINC Atlantic Fleet	SOCPAC	Spec. Op. Cmd-PAC
CINCPACFLT	CINC Pacific Fleet	SOUTHCOM	Southern Command
CNO	Chief of Naval Operations	SPACECOM	US Space Command
EUCOM	European Command	STRATCOM	US Strategic Command
FORSCOM	Forces Command	TRANSCOM	US Trans. Command
HQAF	HQs, US Air Force	USAFE	US Air Force-Europe
HQDA	HQs, Dept of the Army	USASOC	USA Spec. Ops. Cmd
HQMC	HQs, USMC	USFK	US Forces-Korea
JTO	Joint Training Organ.	USFK2	US Forces-Korea (2)

Figure F-1. GCCS IOC Sites

While the basic purpose of all of the GCCS IOC sites is to support the National Command Authorities (NCA), each of the organizations has a specific role in supporting the NCA. In many cases these roles are supported by other organizations (Agencies and/or components) but these relationships are not necessarily overriding factors for including them within the same cell. The administration, security, and overhead considerations can be evaluated based on the current structure of each sites. For the most part, these sites were WWMCCS host sites and will be GCCS JOPES database sites. These sites already have the infrastructure in place to support the administrative and security load for their site. Under WWMCCS, these sites already have site administrators (coordinators) and security officers assigned. They manage administration and security for the entire site.

Our initial cell structure proposal called for 37 GCCS cells based on the 37 IOC sites (Figure F-1). This was done because of the CINC/Service structure already in existence for GCCS. As noted above, the IOC sites already have systems administrators and security officers who perform all of the system administration duties for their sites. In addition, each GCCS site is an individual organization that has an operational relationship but no administrative relationship to the other 37

sites. In order to simplify intercell security and communication, it was proposed that the cells be oriented hierarchically based on the CINC/Service structure. This would allow the use of transitive trust in the hierarchy to reduce intercell authentication activities. It is recommended that cell structure for GCSS and the remaining DII systems also implement a hierarchial cell structure.

When hierarchial cells and transitive is fully implemented, an administrator will be able to establish a trust relationship with one node on the tree and securely communicate with all of the cells below that node. This was done in order to increase system efficiency and does not carry any operational connotation. The main purpose of using hierarchial cells is to avoid having thirty-seven (37) separate cells individually authenticate each other. This hierarchial grouping allows the authentication to take place between only the top levels of the heirarchy with the lower levels being "grandfathered".. For example, if PACOM authenticates TRANSCOM, all of the sites that are grouped under the PACOM hierarchy are authenticated by TRANSCOM

At the current time, hierarchial cells do not work adequately below one level of nesting and transitive trust has not been fully implemented. Indications are that these problems are not going to be resolved in the near term. While our initial proposal is still valid, it would require that the current problems with hierarchial cells be corrected and that transitive trust be fully implemented in the near term. This does not appear to be a likely scenario.

Another concern that impacts on the issue of cell structure is the problem of DCE administrative overhead. Each of the DCE cells would have DCE administration duties that are similar but separate from normal UNIX administration. This includes setting up and maintaining the cell directory service and the security registry. Adding this level of complexity on each of the 37 cells has been a concern.

An analysis of the problems with hierarchial cells and transitive trust have led us to re-evaluate the proposed cell structure and look for alternatives that would lessen the impact of these problems and also address the issue of administrative overhead. The most logical alternative is to reduce the number of cells and flatten out the hierarchy to a single level. This would reduce the number of cells that would have to perform intercell authentication and reduce the administrative overhead. The key to this proposal is to reduce the administrative overhead while not overburdening the cell administrator with the duty of performing the routine user functions (add/delete/modify users) for all of the sites within the cell. This can be accomplished by establishing subdirectories within the registry which will allow local sites to perform the routine user functions just for their sites. Each of the sites would have a subadministrator(s) responsible for their site portion of the directory service and security registry. The overhead that is eliminated under this proposal is the maintenance associated with the Cell Directory Service and the Security registry. Instead of numerous master security registries, there will one master for the cell and

replicas at the sites. The same is true for the Cell Directory Service. Figure F-2 shows (graphically) what the proposed cell structure would look like and what sites will be included in each cell. Instead of each site having to establish separate cell services, the central cell site would establish the services (registries) and provide access to the central registries for the site administrators. The consolidated cell administrator would be responsible for performing backups and routine maintenance on these services/registries.

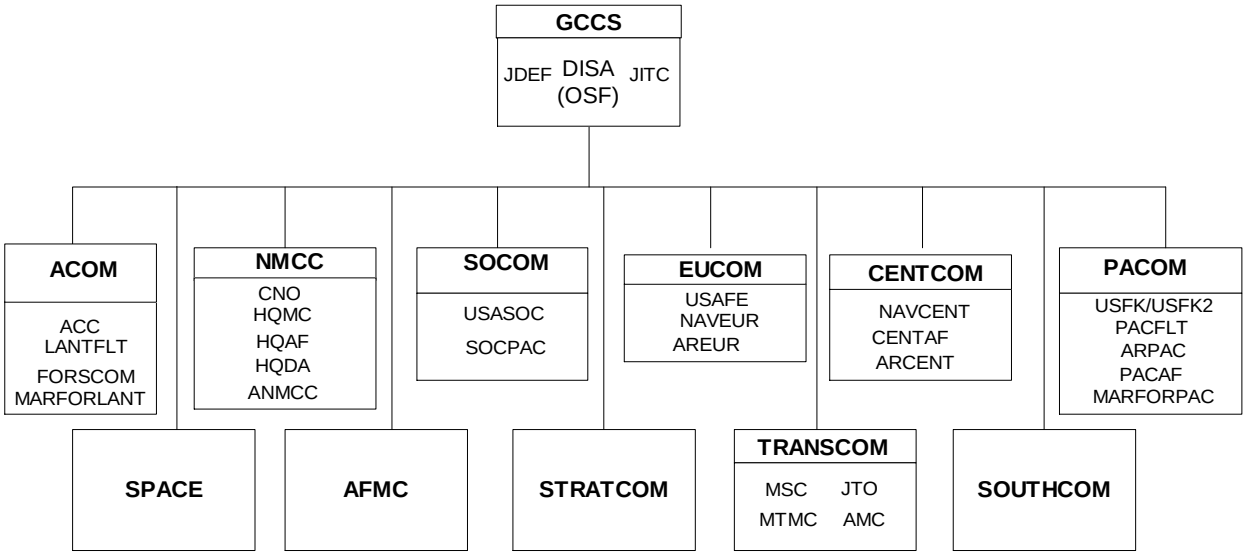

Figure F-2. Proposed Consolidated Cell Structure.

The site personnel would become subadministrators for the cell information that is pertinent to their site and they would have the capability to add/delete/modify user accounts for their site. We are proposing to centralize services while retaining local responsibilities for users. This would eliminate redundant housekeeping chores while retaining individual site responsibilities for user activities (add/delete/modify user).

The following table (Table F-3) provides the same information as figure F-2 but in tabular format.

Table F-3. GCCS Cell Structure.

CELL NAME (DNS DOMAIN NAME)	**SITES /SUB ADMINISTRATORS**
GCCS (.gccs.smil.mil)	JDEF (.jdef.disa.smil.mil)
	JITC (.jitc.disa.smil.mil)

CELL NAME (DNS DOMAIN NAME)	**SITES /SUB ADMINISTRATORS**
ACOM (.acom.smil.mil**)**	ACC (.acc.langley.af.smil.mil)
	CINCLANTFLT (.clf.smil.mil)
	FORSCOM (.forscom.army.smil.mil)
	MARFORLANT (.mfl.usmc.smil.mil)
AFMC (.afmc.wpafb.af.smil.mil)	
CENTCOM (.centcom.smil.mil)	ARCENT
	CENTAF (.centaf.shaw.af.smil.mil)
	NAVCENT-F (.navcent-f.navy.smil.mil)
	NAVCENT-R (.navcent-r.navy.smil.mil)
EUCOM (.eucom.smil.mil)	AREUR (.areur.army.smil.mil)
	NAVEUR (.naveur.navy.mil)
	USAFE (.usafe.ramstein.af.smil.mil)
NMCC (.nmcc.smil.mil)	NMCC-R (.nmcc-r.smil.mil)
	HQDA (.hqda.army.smil.mil)
	HQAF (.hqaf.pentagon.af.smil.mil)
	CNO (cno.navy.smil.mil)
	HQMC (.hqmc.usmc.smil.mil)

CELL NAME (DNS DOMAIN NAME)	**SITES /SUB ADMINISTRATORS**
PACOM (.pacom.smil.mil)	ARPAC (.arpac.army.smil.mil)
	PACAF (.pacaf.hickam.af.smil.mil)
	PACFLT (.pacflt.navy.smil.mil)
	USFK-T (.usfk-t.army.smil.mil)
	USFK-Y(.usfk-y.army.smil.mil)
	MARFORPAC (.mfp.usmc.smil.mil)
SOCOM (.socom.smil.mil)	USASOC (.usasoc.army.smil.mil)
	SOCPAC
SOUTHCOM (.southcom.smil.mil)	
SPACECOM (.spacecom.smil.mil)	
STRATCOM (.stratcom.offutt.af.smil.mil)	
TRANSCOM (.ustc.smil.mil)	MTMC (.mtmc.army.smil.mil)
	AMC (.amc.scott.af.smil.mil)
	MSC (.msc.navy.smil.mil)
	JTO (.jto.scott.af.smil.mil)

Another factor to be considered is the issue of security accreditation. Currently, both WWMCCS and GCCS sites are responsible for certifying the security of their sites in respect to WWMCCS

and GCCS. This means that a Designated Approval Authority (DAA) is appointed for the site/command. In most cases, the Commanding General or Commander of the site is designated the site DAA but delegates that authority down to the J6 or equivalent. It is the DAA who is responsible for reviewing the certification of WWMCCS/GCCS to be operated at that site and certifying that it can be operated in a secure manner in accordance with the appropriate regulations. He also assumes any risk that may be present due to the configuration or operation of the WWMCCS/GCCS equipment. If GCCS sites are consolidated and all security activities were centralized at the cell central site, local sites would lose control of some of their security responsibilities. This would subvert the site certification and accreditation process, and would most likely be unacceptable to the sites. In the new cell structure sites would retain their user responsibilities so the local site still has certification and accreditation responsibilities.

While the new cell structure proposal (figure F-2) provides a reduction in the total number of cells (from 37 to 12), further consolidation can be accomplished. The consolidated cell structure in figure F-2 still maintains a CINC/Service/Agency aligned structure. Further consolidations would erode this alignment. While this is not critical, it may be an issue with the affected organizations. Figure F-4 represents a further consolidation of the structure presented in figure F-2. As shown in figure F-4, any further consolidation would involve aggregating CINC sites like CENTCOM, SOCOM, and SOUTHCOM together into a single cell. This is a logical consolidation since all of the sites are geographically related (all in Florida or soon to be).

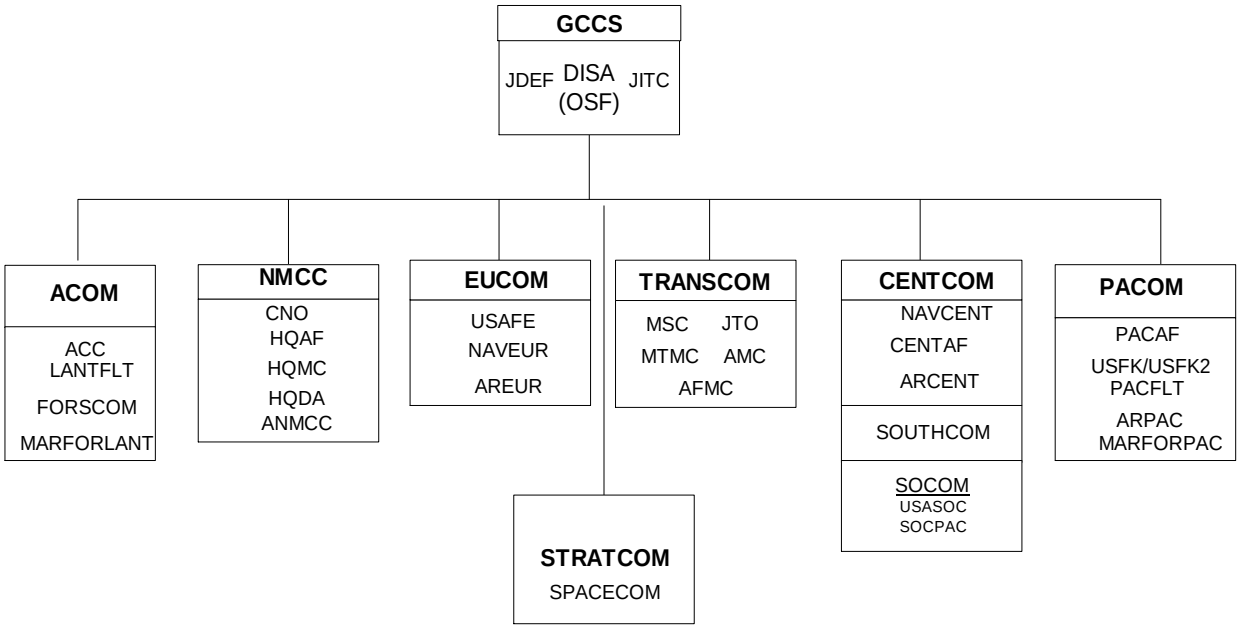

Figure F-4. Further Consolidation.

In addition, SOCOM and CENTCOM are essentially collocated and currently share some resources. The initial consolidation presented in figure F-2 is the favored solution but if further consolidation is required, it is recommended that the cell structure provided in figure F-4 be considered. Table F-5 provides the same information as figure F-4 only in tabular format.

Table F-5. Further Consolidation.

CELL NAME (DNS DOMAIN NAME)	**SITES (SUB ADMINISTRATORS)**
GCCS (.gccs.smil.mil)	JDEF (.jdef.disa.smil.mil)
	JITC (.jitc.disa.smil.mil)
ACOM (.acom.smil.mil**)**	ACC(.acc.langley.af.smil.mil)
	CINCLANTFLT (.clf.smil.mil)
	FORSCOM (.forscom.army.smil.mil)
	MARFORLANT (.mfl.usmc.smil.mil)
CENTCOM (.centcom.smil.mil)	ARCENT
	CENTAF (.centaf.shaw.af.smil.mil)
	NAVCENT-F (.navcent-f.navy.smil.mil)
	NAVCENT-R (.navcent-r.navy.smil.mil)
	SOUTHCOM (.southcom.smil.mil)
	SOCOM (.socom.smil.mil)
	SOCPAC
	USASOC (.usasoc.army.smil.mil)
EUCOM (.eucom.smil.mil)	AREUR (.areur.army.smil.mil)
	NAVEUR (.naveur.navy.mil)
	USAFE usafe.ramstein.af.smil.mil

CELL NAME (DNS DOMAIN NAME)	**SITES (SUB ADMINISTRATORS)**
NMCC (.nmcc.smil.mil)	NMCC-R (.nmcc-r.smil.mil)
	HQDA (.hqda.army.smil.mil)
	HQAF (.hqaf.pentagon.af.smil.mil)
	CNO (cno.navy.smil.mil)
	HQMC (.hqmc.usmc.smil.mil)
PACOM (.pacom.smil.mil)	ARPAC (.arpac.army.smil.mil)
	PACAF (.pacaf.hickam.af.smil.mil)
	PACFLT (.pacflt.navy.smil.mil)
	MARFORPAC (.mfp.usmc.smil.mil)
	USFK-T (.usfk-t.army.smil.mil)
	USFK-Y(.usfk-y.army.smil.mil)
STRATCOM (.stratcom.offutt.af.smil.mil)	SPACECOM (.spacecom.smil.mil)
TRANSCOM (.ustc.smil.mil)	MTMC (.mtmc.army.smil.mil)
	AMC (.amc.scott.af.smil.mil)
	MSC (.msc.navy.smil.mil)
	JTO (.jto.scot.af.smil.mil)
	AFMC (.afmc.wpafb.af.smil.mil)

In addition to the cells created for the IOC sites, it is necessary to create some cells that will make the implementation of DCE more efficient. These additional cells are the supporting sites for DISA and a centralized cell representing the administrative structure necessary to support DCE implementation. The GCCS cell is actually the DISA site at the Operational Support Facility(OSF). The other support cells are the GCCS sites at the Joint Interoperability Test Center (JITC) and the Joint Demonstration and Evaluation Facility (JDEF). The creation of the

DISA cell will facilitate the implementation of the hierarchial cell structure and support the implementation of the Distributed File System (DFS). The DFS servers will be located at the OCONUS Regional Control Centers (RCCs), and at TRANSCOM. The DFS servers will be members of the DISA cell. The new centralized cell is GCCS which includes the Global Control Center which oversees the operation of the SIPRNET and the GCCS Network Operations Center which manages the operations of GCCS.

The proposed cell structure (figure B-2) provides a limited implementation of hierarchial cells. This will be extremely useful if the cell hierarchy and transitive trust problems are corrected in the future. As we move into installing Post-IOC sites, the decision on whether a site should become a member of an existing cell or be an independent cell will depend upon progress in correcting these problems.

F4.0 GCCS Cell Names

All entries in the DCE Directory Service have a global name that is universally meaningful and usable from anywhere in the DCE naming environment. The cell name exists because cells must have names that are accessible from the global naming environment. Cell names are established upon initial configuration of DCE components and is important since all resources of that cell inherit the cell name as part of their own name. Cell names were difficult to change in earlier versions of DCE but they are not difficult to change in version 1.1.

As noted, hierarchial cells and transitive trust have not been fully implemented in DCE 1.1 but these concepts have the potential to significantly reduce overhead in DCE Security administration. Therefore, the GCCS cell structure will be configured as a two level hierarchial structure with hopes that transitive trust will be fully implemented in DCE 1.2.

GCCS cell names will be based on the commonly accepted acronym for CINCs, Service, Agency, unit, or command. The cell name is based on the existing Domain Naming Service used in GCCS.

GCCS hierarchial cell names will be based on the commonly accepted acronym for CINCs/Service/Agency/unit/command. The cell name is based on the existing Domain Naming Service used in GCCS. GCCS uses the Domain Name Service (DNS) for its global name service. Although DNS is not a part of the DCE technology offering, the DCE Directory Service contains support for cells to address each other through DNS. The SIPRNET Support Center (SSC) is responsible for all SIPRNET DNS addressing. It performs the mission that is normally accomplished by the Network Information Center in an unclassified network. The root domain identifier for all secret DOD users is ".SMIL.MIL". The site/host portion of the name will be assigned by the service/agency.

F5.0 Users

GCCS and DII COE end users will notice little difference using DCE in Version 3.0.

Every GCCS user will be assigned a DCE unique user identity. This identity will be used for authorization decisions across all DCE applications and eventually non-DCE components (such as Databases). In order to use a DCE identity, users will have to authenticate to DCE. Hopefully, DCE login can be integrated with UNIX host login and/or Executive Manager login. If not, each user will have to perform a separate DCE login.

Some GCCS users may have a directory in the DFS file space. This directory will be accessible to the GCCS user regardless of their physical location. Users will be free to use this directory as their home directory or as they need. The combination of a global DCE Security identity and file system allows a user to function normally from any GCCS site without the need to perform remote login.

Users that directly perform file transfers will be able to use DFS to transfer files by direct reference (i.e., using UNIX file management commands, file manager, or GCCS Executive Manager foldering system) rather than file transfer protocol (FTP).

F6.0 Hardware

The use of DCE will have no impact on existing GCCS hardware. Server and client resources appear to be sufficient to support DCE.

The four machines comprising the DISA cell will be located at DISA, TRANSCOM, EUCOM, and PACOM. The purpose of this cell is to provide the central cell in the GCCS hierarchy and to support the implementation of DFS. Specifics on the DFS implementation can be found in Appendix D. Each of these machines will be configured similarly to the DII COE server which is currently configured as a SPARC20 with 224 MB of RAM and 16 GB of disk space. Each of these machines will act as a CDS, DFS, and DTS server. The machine in DISA will be the Master Security Server and Initial CDS Server. EUCOM and PACOM will also be configured with Security Replica servers. Eventually, the machine supporting the DCE security server in each cell should be a dedicated and physically secure machine. This is not required for Version 3.0.

F7.0 GCCS DCE Server Allocation

Appendix E provides a generic allocation of DCE servers for the DII cells. The same allocation is applicable to GCCS. In addition, the following table (Table F-6) provides a proposed allocation of servers for the GCCS global cell.

Table F-6. GCCS Global Cell

	dced	secd	cdsd	dtsd	gdad	DFS Gateway	DFS Server	DFS Client
DISA	Y	Master	Initial	Server	Y	N	Y	Y
TRANSCOM	Y	No	Replica	Server	N	N	Y	Y
EUCOM	Y	Replica	Replica	Server	N	N	Y	Y
PACOM	Y	Replica	Replica	Server	N	N	Y	Y

The following table (Table F-7) provides a proposed server allocation for the DFS servers/components for the GCCS Global Cell.

Table F-7. GCCS Global Cell DFS Components

	flserver	ftserver	repserver	fxd	dfsd	butc	dfsbind	bosserver
DISA	Y	Y	Y	Y	Y	Y	Y	Y
TRANSCOM	Y	Y	Y	Y	Y	Y	Y	Y
EUCOM	N	Y	Y	Y	Y	Y	Y	Y
PACOM	N	Y	Y	Y	Y	Y	Y	Y